Pilotierung mobiler Dienste im Automobilsektor

Holger Hoffmann, Jan Marco Leimeister, Helmut Krcmar

Inhaltsverzeichnis

1. Einführung

Telematikdienste im Automobilmarkt waren die letzten Jahre nicht sehr erfolgreich in Deutschland. Als Hauptgründe für deren Scheitern am Markt werden meist die folgenden Gründe genannt:

- die *Kommunikationskosten* waren zu hoch,[1]
- die angebotenen Dienste trafen nicht wirklich die *Bedürfnisse der Kunden,*[2]
- Telematikdienste waren vor allem in der Entwicklung zu sehr auf *Technologien fixiert*,[3] die mangelhafte Berücksichtigung ökonomischer Aspekte machte es fast unmöglich langfristig erfolgreiche Dienste anzubieten.

Durch die Verfügbarkeit neuer, breitbandiger, digitaler Übertragungswege wie UMTS (Universal Mobile Telecommunications System) oder DAB (Digital Audio Broadcast, Digitalradio) und dem Preisverfall, besonders im Mobilfunk, kann das erste der Probleme eliminiert werden. Hieraus ergibt sich die zentrale Forschungsfrage, die in diesem Kapitel adressiert wird: Wie soll eine möglichst Erfolg versprechende systematische Entwicklung innovativer mobiler Dienste für den Automobilsektor erfolgen?

2. Vorgehensmodell zur Entwicklung mobiler Dienste

2.1 Anforderungen an die systematische Entwicklung mobiler Dienste im Fahrzeug

Zur Ermittlung der Anforderungen an mobile Dienste im Fahrzeug sowie deren systematische Entwicklung wurden über zwei Jahre Expertengespräche bei einem bayerischen Automobilhersteller geführt. Die so gesammelten Anforderungen lassen sich wie folgt systematisieren:[4]

---

1 Vgl. Frost/Sullivan (2003).
2 Vgl. Fuhr (2001); Werder (2005).
3 Vgl. Werder (2005).
4 Vgl. hierzu Ehmer (2002).

- *Starke Endkunden-/Nutzerorientierung bei der Diensteentwicklung*: Der Kundennutzen muss kommunizierbar und erfahrbar sein und entsprechend in den Vordergrund gerückt werden. Daraus folgt, dass Rahmenbedingungen für die Erbringung des Dienstes erhoben werden müssen, der Kundennutzen wird in Form von Use Cases abgebildet und Endnutzer werden aktiv in den Erstellungsprozess eingebunden.
- *Dienste sollen nachhaltig profitabel sein*: Aus diesem Grund muss die Zahlungsbereitschaft der Kunden möglichst früh im Entwicklungsprozess annähernd ermittelt werden können, der Business Case des Dienstes mit Value Proposition muss stimmig sein und der Dienst mit seinen Ausbaustufen muss in das Herstellerkonzept integriert werden (können).
- *Dienstleistungen sollen nicht nur alleine, sondern mit Partnern erbracht werden können*: Für jeden Dienst bedarf es eines speziellen Anforderungsprofils für benötigte Partner. Ein akzeptables Geschäftsmodell für jeden Partner muss gefunden werden.
- *Technologie soll die Dienstleistung unterstützen, keine technologiegetriebene Diensteentwicklung*: Technologien müssen auf Rahmenbedingungen des Dienstes gemappt und evaluiert werden können.
- *Dienste müssen sicher während der Fahrt benutzbar sein*: Beachtung der jeweiligen gesetzlichen Vorgaben sowie der eigenen und in der Industrie etablierten Best-Practices je nach Art des Dienstes. Realitätsnahe Prototypenevaluation im Fahrzeug muss möglich sein.

Um mobile Dienste zukünftig erfolgreicher entwickeln zu können, werden im nachfolgend dargestellten Design Framework (vgl. Abbildung 1) diese ökonomischen und technologischen Anforderungen aufgegriffen und zusammengeführt. Ergänzt wird dieses Design Framework durch ein Vorgehensmodell zur effizienten und effektiven Anwendung des Frameworks.

2.2 Das MACS Design Framework

Aufbauend auf den vorhergehend identifizierten Anforderungen ist das MACS Design Framework in folgende Teilbereiche (bzw. Schichten) untergliedert (vgl. MACS

Design Framework), die jeweils einen Schritt im Entstehungszyklus eines mobilen Dienstes beschreiben: Die bereits in den Teilen 1 und 2 beschriebenen Diensteszenarien umfassen den allgemeinen Aufbau und Rahmen, in dem der Dienst zum Einsatz kommen soll, sowie Use Cases für den mobilen Dienst. Anhand der Diensteszenarien werden detaillierte Informationen, wie Nutzeranforderungen oder die Zahlungsbereitschaft (bspw. über Workshops mit Fokusgruppen[5]), für bestimmte Dienste möglichst frühzeitig ermittelt. Eine Analyse der Diensteszenarien nach benötigten Partnern zur Diensteerbringung ergibt Informationen in welcher Form die einzelnen Partner im *Wertschöpfungsnetzwerk* interagieren können und wie das Wertschöpfungsnetz aufgebaut sein kann bzw. soll.

Abbildung 1 MACS Design Framework

Eine der größten Herausforderungen für ein Framework im Automobilbereich sind die unterschiedlich langen Lebenszyklen des Automobils und der im Fahrzeug verwendeten Software.[6] Während die durchschnittliche Lebensdauer eines Autos rund zehn

[5] Vgl. Bortz/Döring (2002).
[6] Vgl. Hartmann (2004).

Jahre beträgt, lösen neue Technologien und Software einander alle zwei bis drei Jahre ab und machen es damit schwierig den Anwendungslebenszyklus und den Automobillebenszyklus zu managen.[7] Um eine tragfähige Auswahl an Technologien und die Erstellung einer Infrastruktur zur Kompensierung des „Lifecycle Mismatch" zwischen Software und Automobil zu ermöglichen, ist besonderes Augenmerk auf die Bereiche *Technologien* sowie *Prototyp und Plattform* zu legen. Hinzu kommt im folgenden Teil 4 die notwendige Betrachtung der *Sicherheitsaspekte* zur Gewährleistung der sicheren Benutzbarkeit des Dienstes während der Fahrt. Der Ausgangspunkt für diese Betrachtungen ist die Verknüpfung der Diensteszenarien mit den verfügbaren Technologien, gefolgt von der Prototypingphase zur Demonstration und Evaluation des Risiko- und Nutzenpotenzials. Ist der Dienst für straßentauglich befunden, folgt die Planung des Roll-Out des *Live Dienstes*.

Die Anwendung solch eines Frameworks zur systematischen, iterativen Entwicklung mobiler Dienste stellt die Entwickler allerdings vor einige Hürden: Strategien für das *Diensteszenario* müssen mit den verfügbaren *Technologien* in Übereinstimmung gebracht werden, das *Wertschöpfungsnetzwerk* muss die Strategien des Szenarios tragen, Beteiligte *unterschiedlicher Arbeitsgebiete* müssen zusammenarbeiten können, d.h. zumindest die Begrifflichkeiten der anderen verstehen und deren Entscheidungen nachvollziehen können, und Funktionalität muss *iterativ aufgebaut* und kontinuierlich anhand bestimmter Faktoren *evaluiert* und *angepasst* werden.

2.3 Vorgehensmodell für die Anwendung des MACS Design Framework

Die Anwendung des MACS Design Frameworks beinhaltet einige Aspekte aus denen sich gezielt Anforderungen für ein Vorgehensmodell der Dienstleistungserstellung ermitteln lassen:

- Die neu konzipierten Dienste stellen *Innovationen* dar. Nicht alle Anforderungen lassen sich deshalb im Vorfeld bestimmen und können sich während der Ent-

[7] Vgl. Mohan (2006).

wicklung auch ändern. Der Entwicklungsprozess muss also *iterativ* abgearbeitet werden können.

- Die mittels des Design Frameworks entwickelten Dienste sollen so gut wie möglich die *Anforderungen der Kunden* treffen. Dies bedingt ein hohes Maß an *Nutzerintegration* in den Entwicklungsprozess.
- Eine valide *Sicherheitsevaluation* für neu entwickelte mobile Dienste ist nur in der vorgesehenen Umgebung, d.h. im Automobil, sinnvoll. Das bedeutet, dass spätestens für die Sicherheitsevaluation ein im Fahrzeug *testbarer Prototyp* verfügbar sein muss.

Aus diesen Anforderungen ergibt sich entsprechend ein iterativer Entwicklungsprozess, der die Anforderungen der Nutzer sowie die Verkehrssicherheit des Dienstes durch Prototypenevaluationen einbeziehen kann. Die in der Informatik oft verwendeten linearen Entwicklungsmodelle sind für diese Anforderungen also nicht anwendbar. Wesentlich geeigneter sind Prozessmodelle wie das Wasserfallmodell[8] oder das Spiralmodell.[9] (Vgl. Abbildung 2) Hier werden die Annahmen eines Prozessschrittes im Modell im folgenden Schritt evaluiert und liefern somit Eingaben für eine neue Iteration. Dies gilt sowohl für weitere Anforderungen während der Analysephase jedes Schrittes (d.h. Informationsfluss von oben nach unten) als auch für Evaluationsergebnisse für vorhergehende Schritte (d.h. Informationsfluss von unten nach oben).

Um den neuen Dienst hinsichtlich der Bedienbarkeit und der Straßentauglichkeit evaluieren zu können, müssen ein gut in die Fahrzeuginfrastruktur integrierter Prototyp erstellt und die Bedienschritte möglichst genau abgebildet werden. Auch hier ist die Anwendung der üblichen Standardmethoden des *vertikalen* oder *horizontalen Prototyping*[10] ineffizient oder kaum zu gebrauchen. Ein horizontaler Ansatz verbietet die Integration des Prototypen in die Infrastruktur des Fahrzeuges, die sinnvolle Evaluation der mobilen Dienste in ihrer späteren Umgebung ist damit unmöglich. Die

---

[8] Vgl. Royce (1970).
[9] Vgl. Boehm (1988b).
[10] Vgl. Floyd (1983).

komplette Integration durch einen vertikalen Ansatz bedeutet andererseits, dass die Benutzung des konkreten Dienstes nicht in seiner Gänze betrachtet werden kann.

Abbildung 2 Spiralmodell nach Boehm[11]

Damit ist wiederum eine Evaluation der Gefährdungspotenzialpotenziale nur bedingt möglich. Diese Probleme löst die Kombination beider Prototypingansätze zu einem diagonalen Prototyping (vgl. Abbildung 3): spezielle oder einzigartige Funktionalität des Dienstes, wie z.B. die Präsentation personalisierbarer Nachrichten oder die Eingabemöglichkeit via Sprachbefehl, wird komplett implementiert (vertikaler Ansatz). Die meisten Funktionen davon in einer Form, wie sie der Kunde im endgültigen Dienst erleben würde, aber nicht notwendigerweise in der gleichen Form, wie sie technisch im finalen Produkt umgesetzt würden (horizontaler Ansatz). Damit wird die Komplexität der Entwicklung in einer hoch proprietären embedded Umgebung (d.h. der Infrastruktur des Autos und seiner Infotainment-Einheit) entschärft. Es ist dabei jedoch immer noch möglich, dem Endnutzer die wichtigen Innovationen in der in einem Fahrzeug erwarteten Form näherungsweise darzustellen.

[11] Vgl. Ebenda.

Abbildung 3 Kombination von horizontalem und vertikalem Prototyping[12]

Kombiniert man das iterativ gestaltete MACS Design Framework mit seinen abgegrenzten Phasen mit dem diagonalen Prototypingansatz ergibt sich das in Abbildung 4 dargestellte MACS-Vorgehensmodell. Im Kern ist ein iterativer Prozess zur Entwicklung, abgeleitet aus den generischen Beschreibungen der Spiralmodelle von Boehm (1988a).[13] Hinzu kommen Elemente aus dem Prototyping. So kann im Vergleich zum ursprünglichen Spiralmodell größeres Augenmerk auf die Ausarbeitung einzelner Szenarien, die Darstellung von (Zwischen-)Ergebnissen in Prototypen und der somit ermöglichten Einbeziehung der Nutzer gelegt werden.

Jede der geplanten Iterationen beginnt mit einer Planungsphase für die Aktivitäten der jeweiligen Iteration. Darauf folgend werden die Anforderungen der aktuellen Phase erhoben, entweder anhand früherer Feldstudien und Experteninterviews (1. Iteration) oder durch die Einbeziehung von Endnutzern und Experten (2. bis 4. Iteration). Die in der Konstruktionsphase erstellten Komponenten/Systeme werden nach Abschluss der Phase evaluiert. Anhand der initialen Anforderungsanalyse und der aus der Evaluation extrahierten sozio-technischen und wirtschaftlichen Anforderungen ergeben sich somit die Eingaben für die folgende Iterationsphase.

[12] Vgl. Ebenda (in Anlehnung).
[13] Vgl. Boehm (1988b).

Abbildung 4 MACS Vorgehensmodell

2.4 Objektorientierte Softwareentwicklung für Automobile

Der konkrete objektorientierte Software-Entwicklungsprozess für mobile Dienste ist als Ausprägung bzw. Instanz des eben vorgestellten Vorgehensmodells der Diensteentwicklung zu verstehen. Die zuvor vorgestellten generischen Einzelschritte von der Anforderungserhebung und Planung zur Evaluation werden in den folgenden Kapiteln für die jeweiligen Entwicklungsphasen konkretisiert.

Im folgenden Kapitel werden die vier entsprechenden Phasen „Anforderungserhebung", „Analyse und Systementwurf", „Implementierung" sowie „Tests und Evaluation" speziell für die technisch orientierten Ebenen und deren Schnittstellen zu den ökonomischen Ebenen näher erläutert. Die vier Phasen zusammen entsprechen dabei einem Iterationsschritt,[14] also dem Durchlaufen einer Spiralwindung.

[14] Vgl. Boehm (1988b).

2.4.1 Anforderungserhebung

Erster Schritt in der jeweiligen Ebene ist die Erhebung der Anforderungen, die das System erfüllen können muss. Endergebnis dieses Schrittes ist eine Menge nicht-funktioneller Anforderungen (ggf. mit einem Use Case Diagramm), funktionaler Anforderungen sowie ein Modell der Abläufe funktioneller Anforderungen.[15]

Es ergeben sich folgende atomare Aktivitäten für die Anforderungserhebung:[16]

1. Identifikation der Akteure (d.h. Nutzer) im System
2. Identifikation von Benutzungsszenarien des Systems
3. Identifikation von Anwendungsfällen und deren Zusammenhängen
4. Identifikation nicht-funktioneller Anforderungen

Bei den zu identifizierenden *Akteuren* handelt es sich im Allgemeinen um Rollen, die von einem oder mehreren Nutzern des Systems belegt sind. Obwohl es sich bei den Akteuren häufig um natürliche Personen handelt, fallen z.B. auch externe Systeme, die mit dem zu erstellenden System kooperieren, unter diese Definition.[17] Mit der Identifikation und Definition der Akteure ist es während der Entwicklung möglich, die Grenzen des zukünftigen Systems abzustecken und die unterschiedlichen Perspektiven, aus denen das System betrachtet wird, festzulegen.[18]

Für eine konkrete, aber freie Beschreibung der einzelnen Anforderungen in der Anforderungserhebung eignet sich die Verwendung von *Szenariobeschreibungen*,[19] da hier narrativ dargestellt wird, welches Verhalten der Kunde vom System erwartet. Allerdings treten hierbei unterschiedliche Probleme auf: So verstehen Kunden die Domäne, aus der die Anforderungen entspringen, haben aber üblicherweise kein Verständnis der Lösungsdomäne. Daraus folgt, dass Anforderungen, die für sie offensichtlich klar erkennbar sind, oftmals nicht vom Kunden kommuniziert werden. Be-

---

[15] Vgl. Brügge/Dutoit (2000), S. 97ff.
[16] Vgl. Jacobson et al. (1992); Jacobson et al. (1999); Wirfs-Brock et al. (1990).
[17] Vgl. Balzert (1998).
[18] Vgl. Brügge/Dutoit (2000), S. 107.
[19] Vgl. Carroll (1995).

reits gesammelte Anforderungen sind auch nicht statisch fix, sondern entwickeln sich laufend, während gemeinsam Szenarien mit dem Kunden erstellt werden.

Als Lösungshilfe für diese Probleme beschreibt Brügge die folgende Heuristik um valide Aussagen zu Szenarien vom Kunden zu erhalten.[20]

1. Anfänglich sind die Szenarien sehr allgemein gehalten und werden iterativ und inkrementell verfeinert.
2. Um Wissen über die Szenarien zu verfeinern, werden bestehende Systeme analysiert und Kunden sowie Endbenutzer, falls möglich, befragt:
   a. Welches sind die Hauptaufgaben, die das System erfüllen können soll?
   b. Welche Daten werden vom Benutzer im System erstellt, gespeichert, verändert, gelöscht oder hinzugefügt?
   c. Welche externen Veränderungen muss das System dem Benutzer mitteilen?
   d. Welche externen Veränderungen muss der Benutzer dem System mitteilen?
3. Visuelle Hilfsmittel wie Mock-Ups der grafischen Benutzerschnittstelle (vgl. Abbildung 4 „Dienstenetzwerk und Technologie“) können helfen, fehlende Anforderungen aufzudecken.

Sind die einzelnen Nutzungsszenarien des Systems bekannt, lassen sich daraus einzelne Anwendungsfälle herleiten. Anwendungsfälle werden von Balzert wie folgt definiert: „Ein [Anwendungsfall] (Use Case) […] besteht aus mehreren zusammenhängenden Aufgaben, die von einem Akteur durchgeführt werden, um ein Ziel zu erreichen bzw. ein gewünschtes Ergebnis zu erstellen“.[21] Jeder Anwendungsfall besteht aus einem beschreibenden Namen und ist mit Akteuren (abstrahierten Benutzern) und weiteren Anwendungsfällen verknüpft. Der Anwendungsfall wird durch einen auslösenden Akteur initiiert,[22] kann aber nur ausgeführt werden, wenn seine Vorbedingungen erfüllt sind. Nach Durchführung der einzelnen Prozessschritte im Anwen-

---

[20] Vgl. Brügge/Dutoit (2000), S. 110.
[21] Balzert (1998), S. 145.
[22] Vgl. Oesterreich (1999), S. 207.

dungsfall gelten spezifizierte Nachbedingungen, die von anderen Anwendungsfällen wiederum als Vorbedingungen genutzt werden können, um Ausführungssequenzen zu beschreiben.[23] Die Ergebnisse dieses Schrittes lassen sich sowohl textuell als auch grafisch, in einem Use Case Diagramm, darstellen.

Nicht funktionelle Anforderungen beschreiben jene Aspekte des fertigen Systems, die zwar nach Außen hin sichtbar sind, jedoch nicht direkt mit der Funktionalität des Systems zusammenhängen.[24] Beispiele für nicht funktionelle Anforderungen sind u.a. Vereinbarungen zur Dokumentation und Performance oder mögliche zukünftige Ausbaustufen. Besonders wichtig für mobile Dienste im Fahrzeug sind nicht-funktionelle Anforderungen (z.B. Gestaltung des Benutzerinterfaces; sichere Bedienbarkeit des Systems während der Fahrt; Hardwareinfrastruktur, auf der der Dienst später laufen soll; externe Einflussfaktoren, z.B. der Temperaturbereich, in dem das System funktionieren muss – der übliche Bereich im Automotive Bereich ist von -40°C bis +85°C).

Ergebnis der Anforderungserhebung ist die schriftliche Fixierung der Rahmenbedingungen, in denen ein System zu funktionieren hat, sowie dessen Verhalten zur Laufzeit.

2.4.2 Analyse und Systementwurf

Aus diesen Informationen werden im folgenden Schritt, der Analyse und dem Systementwurf, die Bedingungen, die das System erfüllen muss (deskriptiv und aus der Problemdomäne), formalisiert in die Lösungsdomäne übertragen[25] und ein komplettes Komponentenmodell der dem System zu Grunde liegenden Software erstellt.[26] Das Ergebnis dieses Schrittes sind Analysemodelle der einzelnen Objekte bzw. ein Analysemodell der Systemdynamik sowie Entwurfsmodelle der jeweiligen Objekte und der einzelnen Subsysteme.

---

[23] Vgl. Balzert (1998), S. 127f.; Jacobson et al. (1992).
[24] Vgl. Brügge/Dutoit (2000), S. 188f.
[25] Vgl. Balzert (1998), S. 376ff.
[26] Vgl. Brügge/Dutoit (2000), S. 167.

In der **Analysephase** werden die erhobenen Anforderungen aus der ersten Phase, die in Form von Use-Case-Beschreibungen vorliegen, neu strukturiert.[27] Nach der Restrukturierung stellen sie noch immer das System aus Anwendersicht dar, allerdings werden bereits einzelne Komponenten bzw. Objekte des Systems identifiziert.[28] Brügge beschreibt die nötigen Schritte sehr detailliert, um aus den Use Cases und Szenarien der Anforderungserhebung ein komplettes Analysemodell, bestehend aus funktionalem Modell (Use Cases und Szenarien), Objektmodell (Klassen- und Objektdiagramme) und dynamischem Modell (Zustands- und Sequenzdiagramme), herzuleiten.[29]. Im Wesentlichen handelt es sich um die semantische Bearbeitung der Anforderungserhebung, so gibt Abbott Regeln vor, mit deren Hilfe sich natürlichsprachige Beschreibungen kategorisieren und auf Modellkomponenten mappen lassen.[30] Das Resultat der Analyse ist ein Modell, welches das System komplett aus Benutzersicht beschreibt. Es beinhaltet keinerlei Informationen bezüglich der Systeminterna oder der Hardwarestruktur.

Um diese Punkte zu ermitteln, wird im **Systementwurf** aus den eben erhobenen Anforderungen ein komplettes Komponentenmodell der dem System zu Grunde liegenden Software erstellt.[31] Dies beinhaltet die Definition von Entwurfszielen sowie eine Beschreibung der Softwarearchitektur.[32] Die Entwurfsziele leiten sich aus den nicht funktionalen Anforderungen ab und dienen zur Entscheidungsfindung bei Problemen während der Umsetzung der Anforderungen in das Endsystem. Den wesentlich größeren Anteil am Systementwurf hat die Beschreibung der Softwarearchitektur. In ihr wird das System in einzelne Systemkomponenten zerlegt, um kleine, leicht handhabbare Systemkomponenten zu erhalten.[33] Dabei finden besonders folgende Punkte Beachtung:

[27] Vgl. Jacobson et al. (1992)1992.
[28] Vgl. Balzert (1998), S. 376ff.; Booch (1994); Coad/Yourdon (1991a); Rumbaugh et al. (1991); Shlaer/Mellor (1988).
[29] Vgl. Brügge/Dutoit (2000), S. 133ff.
[30] Vgl. Abbott (1983).
[31] Vgl. Shlaer/Mellor (1992); Booch (1991); Coad/Yourdon (1991b).
[32] Vgl. Brügge/Dutoit (2000), S. 170.
[33] Vgl. Ebenda.

- *Hardware/Software Mapping*: Wie sieht die Hardwareplattform aus, auf der das System laufen wird? Welche Einzelkomponenten gibt es und wie kommunizieren diese untereinander?
- *Datenhaltung*: Welche Daten müssen dauerhaft gespeichert werden? Wie werden sie gespeichert und wie wird auf sie zugegriffen?
- *Zugriffskontrolle*: Wer darf auf gespeicherte Daten zugreifen? Wie wird die Zugriffsberechtigung festgelegt und umgesetzt?
- *Kontrollfluss*: Wie läuft der Programmfluss ab? Welche Komponenten haben Vorrang?
- *Randbedingungen*: Wie wird das System initialisiert? Wie wird es beendet? Wie wird mit Fehlerfällen umgegangen?

Ergebnis von Analyse und Systementwurf sind die komponentenbasierte Kundensicht auf das System (Analyseergebnis) sowie ein in die Teilsysteme zerlegtes Komponentenmodell des Systems (Ergebnis des Systementwurfes).

2.4.3 Implementierung

In den beiden vorangegangenen Schritten der Softwareentwicklung wurden sowohl der Zweck des Systems (Anforderungserhebung), als auch der Aufbau des Systems erstellt (Analyse und Systementwurf). In der Implementierungsphase werden nun die Konzepte des Systementwurfes in einzelne, ausführbare Komponenten überführt. Brügge schlägt dafür folgendes Vorgehen vor.[34]

- *Dienstespezifikation*: Klassen/Objektinterfaces werden einzeln beschrieben.
- *Komponentenauswahl*: Fertig verfügbare „Off-the-Shelf“ Komponenten werden identifiziert.
- *Objektmodellrestrukturierung und -optimierung*: Das Objektmodell wird zur besseren Verständlichkeit und Erweiterbarkeit umgestellt und auf Laufzeitverhalten, Speicherbedarf o.ä. hin optimiert.
- Die *Implementierung* selbst bildet das letzte Arbeitspaket in diesem Schritt.

---

[34] Vgl. Ebenda, S. 231.

Während der **Dienstespezifikation** werden die einzelnen Objekte genau festgelegt.[35] Dabei wird unterschieden zwischen Anwendungsobjekten, die ein Konzept aus der Anwendungsdomäne beschreiben, und Lösungsobjekten, die als Hilfskonstrukte lediglich in der Lösungsdomäne zu finden sind.[36] Um ein Objekt zu beschreiben, ist es nötig, programmiersprachenabhängige Faktoren wie Typen von Rückgabewerten, Funktionssignaturen (d.h. die Anzahl und Typen der Funktionsparameter) und vor allem auch die Sichtbarkeit (d.h. „Zugriffsberechtigungen") der Funktionen festzulegen.[37] Über Randbedingungen werden schließlich sichere Annahmen über das Verhalten der Objekte definiert.[38] Über *Invarianten* werden Eigenschaften festgelegt, die zu jeder Zeit und für jede Instanz des Objektes zutreffen. *Vorbedingungen* (Preconditions) definieren Eigenschaften, die vor Aufruf einer Methode des Objektes vom Aufrufenden erfüllt sein müssen. Analog dazu bestimmen *Nachbedingungen* (Postconditions) Eigenschaften, die nach Ausführung einer Operation zutreffend sind.[39] Randbedingungen können in Unified Modeling Language-Diagrammen (UML) zu den Klassen mit Hilfe der UML Object Constraint Language dargestellt werden.

Ziel der **Komponentenauswahl** ist es, möglichst redundante Arbeit zu vermeiden, indem bereits existierende Lösungen wieder verwendet werden. Dabei werden die Anforderungen aus der Erhebungs- bzw. Analysephase mit den angebotenen „Off-the-Shelf" Komponenten von Drittanbietern abgeglichen und diese gegebenenfalls angepasst.[40] Eine besondere Rolle bei den vorgefertigten Komponenten spielen Anwendungsframeworks. Sie stellen wieder verwendbare Teilapplikationen dar, die zur Erstellung von eigenen Anwendungen verwendet werden können.[41] Dabei ist die Erweiterbarkeit durch Hook Methoden, welche es erlauben, eigene Änderungen zum bestehenden Code hinzuzufügen, von großer Bedeutung.[42] Anwendungsframeworks lassen sich nach deren Bedeutung im Entwicklungsprozess grob in drei Arten unterteilen:

---

[35] Vgl. Ebenda, S. 167; Booch (1994).
[36] Vgl. Brügge/Dutoit (2000), S. 237.
[37] Vgl. Balzert (1998), S. 814ff.
[38] Vgl. Meyer (1997).
[39] Vgl. Brügge/Dutoit (2000), S. 238f.
[40] Vgl. Ebenda, S. 253ff.
[41] Vgl. Johnson/Foote (1988).
[42] Vgl. Brügge/Dutoit (2000), S.255.

- *Infrastrukturframeworks* dienen der Vereinfachung des Software-Entwicklungsprozesses und sind üblicherweise nicht Bestandteil des fertigen Systems.
- *Middleware Frameworks* werden für komponentenbasierte Systeme, vor allem im Bereich des Lebenszyklusmanagement und der Kommunikation, verwendet.
- *Enterprise Application Frameworks* sind auf die Anwendungsdomäne, z.B. Telekommunikation oder Automotive Infotainment, zugeschnitten.

Diese Anforderungen lassen sich im letzten Schritt der Implementierung (vor dem Umsetzen in den reinen Code), der **Objektmodellrestrukturierung und -optimierung**, umsetzen. Bei der Restrukturierung wird vor allem darauf geachtet, die Wiederverwendbarkeit einzelner Komponenten sicherzustellen und Abhängigkeiten von konkreten Implementierungen aufzuheben.[43] Da die direkte Umsetzung des Analysemodells aus dem vorhergehenden Schritt oft zu ineffizienten Objektmodellen führt,[44] ist es nötig, das erhaltene Objektmodell hinsichtlich der in der Anforderungserhebung ermittelten Werte, z.B. zu Laufzeitverhalten oder Speicherbedarf, zu optimieren. Dazu gehört neben dem Überarbeiten bestehender Objekte in erster Linie auch die Planung rechenintensiver Operationen.[45] In der anschließenden **Implementierung** wird das Objektmodell in der ausgewählten Programmiersprache umgesetzt und zu einem lauffähigen Programm ausgebaut. Auf Basis dieses Programms können dann im folgenden Schritt Tests durchgeführt und das System evaluiert werden.

2.4.4 Test & Evaluation

Das Erstellen neuer Software ist ein kreativer und innovativer Prozess. Bei der Suche nach Lösungen zu den auftretenden Problemen werden alte Denkweisen ständig durch neues Wissen und Rekombination abgelöst.[46] Aus diesem Grund ist es wichtig, die entstandene Lösung dahingehend zu überprüfen, in wie weit sie sich vom geplanten und erwarteten Verhalten unterscheidet.[47] Von besonderer Bedeutung für innovative Dienste sind dabei Evaluationen durch den Endnutzer des Systems. Bei

---

[43] Vgl. Ebenda, S. 259ff. und S. 267ff.
[44] Vgl. Ebenda, S. 270.
[45] Vgl. Ebenda, S. 271f.
[46] Vgl. Grams (1990), S. 13.
[47] Vgl. Brügge/Dutoit (2000), S. 327.

diesen Evaluationen, auch Feldtest oder Pilottest genannt, ist das Feedback der Kunden besonders nützlich, da in Innovationsprojekten anders keine genauen Anforderungen für den Endkunden erhoben werden können.[48] Passend zu den unterschiedlichen Zielsetzungen der einzelnen Entstehungsphasen, „Anforderungserhebung“, „Analyse und Systementwurf“ sowie „Implementierung“, existieren für jede der Phasen auch passende Tests auf Zielerreichung. (Vgl. Abbildung 5)

Evaluation

Anforderungs-erhebung ⇒ Funktionstests Pilottests

Analyse & Systementwurf ⇒ System- & Modultests

Implementierung ⇒ Unit Tests

Abbildung 5 Softwaretests in den einzelnen Iterationsphasen

Während der *Implementierung* stehen dem Entwickler vor allem Unit Tests und Integrations Tests zur Verfügung, um die Korrektheit der Arbeit zu überprüfen, sowie funktionelle Tests, um die Erfüllung funktionaler Anforderungen zu gewährleisten.[49] Unit Tests beziehen sich dabei auf die einzelnen Bausteine des Systems, also Objekte und Subsysteme, und sollten möglichst häufig durchgeführt werden, um frühzeitig Fehler zu finden und beheben zu können.[50] Aus diesem Grund sind ständige Tests fester Bestandteil von Ansätzen wie Extreme Programming[51] und Continuous Integration.[52] Integrations Tests befassen sich mit Gruppen von Objekten und Subsystemen, die nach und nach um weitere Objekte und Subsysteme erweitert werden.

48 Vgl. Ebenda, S. 362.
49 Vgl. Brügge/Dutoit (2000), S. 334f.
50 Vgl. Ebenda, S. 345.
51 Vgl. Jeffries et al. (2001), S. 117 ff.
52 Vgl. Fowler (2006).

Dadurch werden Fehler sichtbar, die erst durch das Zusammenspiel mehrerer Komponenten entstehen, für Unit Tests also nicht erkennbar sind.[53]

Auf der nächst höheren Abstraktionsebene, *Analyse und Systementwurf*, steht nicht mehr die konkrete Umsetzung in Form von Objekten an sich auf dem Prüfstand, sondern vielmehr das System in seiner Gesamtheit.[54] Bei System- und Modultests wird überprüft, ob das System die erhobenen funktionalen Anforderungen erfüllt und ob die nicht-funktionalen Anforderungen eingehalten werden. Brügge und Dutoit[55] schlagen für Systemtests eine Black-Box-Technik vor. Dabei werden aus den Use Cases der Anforderungserhebung Testfälle hergeleitet und im System durchgespielt. Da vor allem bei großen Softwareprojekten nicht alle Use Cases als Testfälle benutzt werden können, werden ausschließlich jene Fälle betrachtet die von hoher Relevanz oder sehr fehleranfällig sind. Durch derartige Black-Box-Tests lassen sich die Unterschiede im geforderten Verhalten, aus der Anforderungserhebung, zum tatsächlichen Verhalten, den Ergebnissen der Testfälle, erkennen und somit die Umsetzung überprüfen. Um die Erfüllung der nicht-funktionellen Anforderungen zu überprüfen, gibt es eine Reihe von Werkzeugen: So lässt sich mittels Stress- und Volumentests das Laufzeitverhalten des Systems unter Extrembedingungen testen, gezielte Securitytests dienen dazu, die Sicherheit des Systems zu überprüfen und Recovery Tests helfen, die Wiederaufnahme des geregelten Betriebes nach einem Fehlerfall zu testen.[56]

Die Güte der *Anforderungserhebung* wird schließlich mit Hilfe von Funktionstests überprüft.[57] Dazu wird das fertige System einem ausgewählten Nutzerkreis vorgestellt und von den Nutzern einem der alltäglichen Nutzung entsprechenden Test unterzogen (Black-Box-Technik). Alternativ können Pilottests zur Nutzerakzeptanz des Systems, auch unabhängig von der Anforderungserhebung, verwendet werden. Man unterscheidet die Pilottests dabei in α-Tests, bei denen die Nutzer das System in der Entwicklungsumgebung ausführen, und β-Tests, bei denen das System in der

[53] Vgl. Brügge/Dutoit (2000), S. 352f.
[54] Vgl. Ebenda, S. 334.
[55] Vgl. Ebenda, S. 359.
[56] Vgl. Ebenda, S. 361f.
[57] Vgl. Ebenda, S. 359.

Zielumgebung, z.B. dem Automobil, ausgeführt wird.[58] Vor jedem Pilottest muss festgehalten werden, welche Funktionen des Systems überprüft werden sollen und wie die Ergebnisse des Tests als Ergebnisse in eine weitere Iteration einfließen.[59]

Neben der Qualitätssicherung durch Produkttests und Produktevaluationen ist ebenfalls die Sicherung des Erstellungsprozesses von Bedeutung.[60] Einer der für das hier vorgestellte Vorgehensmodell zur Software-Entwicklung für mobile Dienste im Fahrzeug wichtigsten Punkte ist die Beachtung dessen iterativer Natur. Beispielsweise gibt es in iterativen Projekten keine klassischen Meilensteine zur Bewertung des Projektfortschritts vor dem Kunden, Stevens schlägt als Beispiel die Übergabe einer Iteration, die über bestimmte Funktionalität verfügt, vor.[61] Als zusätzliche Sicherung im Prozess komponentenbasierter Entwicklungen (vgl. Kapitel 3.1) sollten mögliche Komponenten ermittelt sowie analysiert und das Ergebnis dokumentiert werden, geeignete Komponenten zusätzlich getestet und gegebenenfalls in die Komponentenbibliothek aufgenommen werden.[62]

3. Praktische Anwendung des MACS Design Framework

Im folgenden Kapitel wird die Anwendung der Ebenen „Technologie“ sowie „Prototyp und Plattform“ des MACS Design Framework (siehe Abbildung 1) auf Basis der in Kapitel 2.4 beschriebenen, objektorientierten Entwicklungsmethodik angewandt, um einen exemplarischen Dienst prototypisch für eine Sicherheitsevaluation (siehe Teil 4) umzusetzen und eine Plattform für mobile Dienste bereitzustellen. Dabei erfolgen die Anforderungserhebungen und -analyse sowie der Systementwurf für beide Teile, Diensteplattform wie auch Dienst, gemeinsam mit dem Industriepartner. Nach dem diagonalen Prototypingansatz (vgl. Kapitel 2.3) ausgewählte Anforderungen wurden für eine anschließende Evaluierung des Diensteszenarios sowie eine Sicherheitsevaluation der Dienstenutzung im Fahrzeug ausgewählt und umgesetzt.

---

[58] Vgl. Ebenda, S. 362.
[59] Vgl. Oesterreich (1999), S. 111.
[60] Vgl. Stevens/Pooley (2000), S. 266.
[61] Vgl. Ebenda, S. 280f.
[62] Vgl. Ebenda, S. 282.

3.1 MACS Diensteplattform

3.1.1 Zieldefinition

Eine Zielsetzung des Projekts MACS ist die Beschreibung einer Diensteplattform, mit der sich mobile Breitbanddienste beliebiger Serviceprovider in verschiedenen Endgeräten nutzen lassen, damit ein möglichst breites Spektrum an Diensten (vgl. Abbildung 15) umsetzbar ist. Durch die Nutzung existierender Standards und deren Erweiterung um zusätzliche – öffentlich verfügbare – Standards soll die Unabhängigkeit von einzelnen Softwarezulieferern und von spezieller Hardware gewährleistet werden. Als Basis für die MACS Diensteplattform dient der OSGi Standard. In diesem Kapitel werden Basisdienste, die von dem prototypisch umzusetzenden Dienst MACS MyNews genutzt werden, sowie einige allgemeine Basisdienste beschrieben.

3.1.2 Anforderungserhebung

Bei der MACS Dienstleistungsplattform handelt es sich um ein Enterprise Application Framework mit Middlewarecharakter (vgl. 2.4.3). Entsprechend gibt es mehrere Akteure, die mit der Plattform interagieren. Zum einen sind dies natürliche Personen, wie der Dienstenutzer (Fahrer, Nutzer) oder ein Mitarbeiter einer Vertragswerkstatt (Administrator). Zum anderen interagieren auch weitere Systeme mit der Plattform, allen voran Dienste, die auf der Plattform basieren, und Sensoreinheiten im Fahrzeug. Zu den möglichen Sensoren zählen z.B. Sensoren, die den Zustand des Fahrzeuges überprüfen (wie Motordrehzahl, Öltemperatur oder Benzinstand), aber auch Sensoren, die Umweltinformationen erfassen (z.B. Regen- oder Lichtsensor) oder auf die Fahrsituation spezialisierte Sensorkomponenten, wie ein Lane-Departure-Warning System.

1. ***Anwendungsfall: Umwelt-/Fahrzeuginformationen eingeben***

Akteure: Sensoreinheit

Beschreibung: Eine Sensoreinheit gibt nach einer Änderung ihren aktuellen Messwert über die Umwelt (z.B. Regenmenge, Temperatur) oder das Fahrzeug (z.B. Reifendruck, Lane Departure Warning) betreffend weiter. Aus allen vorliegenden Informationen wird dann das aktuelle Safety Level berechnet.

2. Anwendungsfall: Safety-Level abfragen

Akteure: Interfacesteuerung (Dienst)

Beschreibung: Die Interfacesteuerung fragt Informationen zur aktuellen Fahrsituation ab, in der sich der Fahrer befindet. Entsprechend der Rückgabeinformation gibt die Interfacesteuerung Informationen angepasst an die Fahrsituation, ggf. gar nicht, aus.

Abbildung 6 Anwendungsfälle 1 und 2

3. Anwendungsfall: Information auf Display anzeigen

Akteure: Dienst

Vorbedingungen: Fahrzeug langsamer als 5km/h

Beschreibung: Ein Dienst kann Informationen auf dem Display darstellen. Die grafische Darstellung von Inhalten ist während der Fahrt (definiert als Geschwindigkeit > 5 km/h) nicht zulässig. Menge und Umfang der dargestellten Information sind von der Fahrsituation abhängig.

4. Anwendungsfall: Audioausgabe von Information

Akteure: Dienst

Beschreibung: Ein Dienst kann Informationen über die Fahrzeuglautsprecher ausgeben. Es macht dabei keinen Unterschied, ob es sich um Audiodaten handelt oder um Texte, die von einem Sprachsynthesesystem vorgelesen werden. Art und Umfang der ausgegebenen Information sind von der Fahrsituation abhängig und werden jeweils bei Kollisionen anhand definierter Regeln nach Relevanz priorisiert.

Abbildung 7 Anwendungsfälle 3 und 4

5. Anwendungsfall: Haptische Befehlseingabe

Akteure: Fahrer, Nutzer

Beschreibung: Der Fahrer oder Nutzer kann Befehle über ein haptisches Interface, z.B. über einen Touchscreen, eine Bedieneinheit in der Mittelkonsole oder am Lenkrad, eingeben. Bestimmte Eingaben sind länderspezifisch, während der Fahrt gesperrt oder eingeschränkt möglich, z.B. Programmierung des Navigationssystems.

6. Anwendungsfall: Sprachbedienung

Akteure: Fahrer, Nutzer

Beschreibung: Der Fahrer oder Nutzer kann auch per Sprache mit den Diensten interagieren. Dazu kann es notwendig sein, vor Beginn der Spracherkennung eine Taste am Lenkrad (Push-to-Talk Button, PTT) zu drücken.

Sonderfälle: Wird ein Sprachbefehl nicht erkannt, erfolgt direkt eine akustische Fehlermeldung.

Abbildung 8 Anwendungsfälle 5 und 6

7. Anwendungsfall: SIM Access Profil Status abfragen

Akteure: Dienst

Beschreibung: Dienste haben die Möglichkeit, den Status eines SIM Access Profile abzufragen. Dies ist gleichbedeutend mit einer Abfrage der Verfügbarkeit eines Mobiltelefons. Die Verfügbarkeit des SIM Access Profile gibt jedoch keinen Anhaltspunkt, ob eine Netzverbindung aufgebaut werden kann.

8. Anwendungsfall: SIM Access Profile anfordern

Akteure: Dienst

Vorbedingung: Mobiltelefon verfügbar

Beschreibung: Ist ein Mobiltelefon verfügbar, so können Dienste auf die Daten der SIM Karte im Telefon zugreifen. Beim Datentransfer übernimmt die Diensteplattform die Rolle des SIM Access Client (d.h. des zugreifenden Geräts) und initiiert die Verbindung zum SIM Access Server (d.h. demKartenlesegerät), dem Mobiltelefon. Nachdem der Benutzer (z.B. durch Eingabe seiner PIN) die Verbindung autorisiert hat, kann der Client die SIM Karte im Server komplett auslesen und somit die Identität übernehmen. Da die SIM Karten als einzigartige Benutzer IDs gegenüber den

Mobilfunkbetreibern dienen, bietet sich so die Möglichkeit die Identität eines Benutzers anhand der Daten, die von seiner SIM Karte ausgelesen wurden, festzustellen.

Sonderfälle:

- Wird die Verbindung zwischen Plattform und Mobiltelefon getrennt, wird automatisch versucht, eine neue Verbindung aufzubauen.
- Schlägt die Autorisation fehl (z.B. aufgrund einer falschen PIN), wird eine Fehlermeldung ausgegeben.

9. Anwendungsfall: Netzstatus prüfen

Akteure: Dienst

Vorbedingung: Mobiltelefon verfügbar

Beschreibung: Dienste können die möglichen Verbindungstypen (z.B. GSM, GPRS, UMTS) sowie den jeweiligen aktuellen Verbindungsstatus eines Mobiltelefones abfragen.

Abbildung 9 Anwendungsfälle 7, 8 und 9

10. Anwendungsfall: Daten speichern

Akteure: Dienst

Beschreibung: Dienste haben die Möglichkeit, Daten zur persistenten Sicherung zu hinterlegen.

Sonderfall: Ist kein Speicherplatz im System vorhanden, erfolgt eine Fehlermeldung

11. Anwendungsfall: Daten laden

Akteure: Dienst

Beschreibung: Dienste können Daten direkt anfordern, wobei die Quelle der Daten dabei vernachlässigt werden kann.

12. Anwendungsfall: Daten aus persistentem Speicher laden

Akteure: Dienst

Übergeordneter Anwendungsfall: Daten laden

Beschreibung: Analog zum Anwendungsfall 10 haben Dienste die Möglichkeit, Daten aus dem persistenten Speicher zu laden.

Sonderfall: Sind direkt angeforderte Daten nicht verfügbar, wird eine Fehlermeldung ausgegeben.

13. Anwendungsfall: Daten von Broadcastmedium laden

Akteure: Dienst

Übergeordneter Anwendungsfall: Daten laden

Beschreibung: Dienste können Daten anfordern, die über ein Broadcastmedium (z.B. digitales Radio, DAB), übertragen werden.

Sonderfall: Ist eine direkt angeforderte Datei (noch) nicht verfügbar, wird eine Fehlermeldung ausgegeben.

14. Anwendungsfall: Daten über direkte Verbindung laden

Akteure: Dienst

Übergeordneter Anwendungsfall: Daten laden

Beschreibung: Dienste können Daten über eine direkte Verbindung (z.B. GSM, IP-basiert) herunterladen.

Sonderfälle: Ist eine direkt angeforderte Datei nicht verfügbar, wird eine Fehlermeldung ausgegeben. Kann die gewünschte direkte Verbindung nicht hergestellt werden, wird eine Fehlermeldung ausgegeben.

Abbildung 10 Anwendungsfälle 10 bis 14

15. Anwendungsfall: Update der Diensteplattform

Akteure: Administrator

Beschreibung: Die einzelnen Komponenten der Diensteplattform können von einem Administrator, z.B. in einer Vertragswerkstatt, aktualisiert werden.

16. Anwendungsfall: Installation neuer Plattformkomponenten

Akteure: Administrator

Beschreibung: Neue Komponenten der Diensteplattform können von einem Administrator, z.B. in einer Vertragswerkstatt, installiert werden.

Abbildung 11 Anwendungsfälle 15 und 16

17. Anwendungsfall: Installation/Update neuer Dienste

Akteure: Administrator

Beschreibung: Administratoren, z.B. eine Vertragswerkstatt, können neue Dienste, die auf der Diensteplattform basieren, installieren. Ebenso ist es möglich, bereits im System vorhandene Dienste auf die aktuelle Version upzudaten.

Abbildung 12 Anwendungsfall 17

3.1.3 Analyse und Systementwurf

Eine gute Abstraktion der Abläufe in einem Framework für die Entwicklung einer Plattform zu finden, ist eine Aufgabe, die mehrere Iterationsschritte umfasst.[63] Um ein zuverlässiges Design in möglichst wenigen Iterationsschritten zu erhalten, ist es nötig, sowohl kontextspezifische Anforderungen zu erheben, als auch die aktuellen Best Practices in einem möglichst ähnlichen Gebiet zu analysieren. Aus diesem Grund kommt eine Kombination zweier Analysemethoden zum Einsatz. Die kontextspezifische Analyse der Anforderungen in der Domäne (d.h. mobile Dienste im Fahrzeug) werden mittels der „Domain Analysis“[64] durchgeführt und durch eine Best-Practice-Analyse[65] der beiden am Markt zur Verfügung stehenden Architekturen, der Siemens “Top Level Architecture” (TLA) und von AutoSAR[66], komplettiert.

Um eine strikt getrennte modulare Infrastruktur im Fahrzeug zu schaffen, wird die Funktionalität des MACS Framework in „Basisdienste“ aufgeteilt, in denen Funktionen logisch gruppiert sind und die untereinander kommunizieren können (vgl.

63 Vgl. Fayad et al. (1999).
64 Vgl. Aksit et al. (1999).
65 Vgl. Boone (1999).
66 Vgl. Heinecke et al. (2004).

Abbildung 13). Als Laufzeitumgebung und als Kommunikationsplattform kommt das häufig im Automotive Sektor eingesetzte OSGi Framework[67] zum Einsatz. Jeder dieser Basisdienste steht entweder für ein Interface zur Infrastruktur des Automobils, stellt die sichere Benutzbarkeit (auch während der Fahrt) der mobilen Dienste sicher oder ist ein weiterer Architekturbaustein in OSGi.[68] Das OSGi Framework selbst stellt dabei wichtige und robuste Funktionen bereits zur Verfügung, u.a. das Lebenszyklusmanagement für Komponenten und das dynamische Finden von Funktionalität. Darüber hinaus ist es möglich, das OSGi-basierte System aus der Ferne zu aktualisieren oder neue Komponenten hinzuzufügen.[69]

Dienste (e.g. MACS MyNews)
Basisdienste
OSGi Framework
Betriebssystem + Java Virtual Machine

Abbildung 13 Architekturübersicht der MACS Diensteplattform

3.1.4 Implementierung

Basierend auf der Anforderungserhebung aus Kapitel 3.1.2 und deren Analyse gekoppelt mit dem Systementwurf aus dem Kapitel 3.1.3 wurden für das Pilotsystem der MACS Diensteplattform Funktionen ausgewählt, die für das Anwendungsszenario durch MACS MyNews (vgl. Kapitel 3.2) oder für die Evaluationsphase (vgl. Teil 6) von besonderer Bedeutung sind. Dabei wurden – so weit wie möglich – bestehende Komponenten „Off-the-Shelf" verwendet, um für eine mögliche spätere Umsetzung und die mögliche Übertragbarkeit der Ergebnisse realitätsnahe Szenarien zu gestalten. (Vgl. Abbildung 14)

---

[67] Vgl. Alliance (2005).
[68] Vgl. Ebenda.
[69] Vgl. Wong (2001); Palenchar (2002).

MACS Diensteplattform
Application Safety Manager
Input Manager
Software Manager
Output Manager
Daten- und Persistenzmanager

Abbildung 14 Komponentenübersicht MACS Diensteplattform Pilotsystem

1. Application Safety Manager

Der Application Safety Manager stellt Funktionen zur Gewährleistung der Sicherheit des Fahrers zur Verfügung. Auf Basis aktueller Situationsdaten werden dazu „Safety Levels“ berechnet, anhand derer die Interaktion des Dienstenutzers mit dem Dienst geregelt wird. Dabei dient der Application Safety Manager als Informationsgradakzeptor für Ausgaben, die von anderen Basisdiensten oder Diensten angefordert werden. Um das aktuell akzeptable Maß an Information in Menge und Format festzusetzen, werden durch den Application Safety Manager die Informationen der verfügbaren Fahrzustandsrezeptoren aggregiert und bewertet.

Die Bewertungsfunktion mit den Gewichtungen der einzelnen Rezeptorinformationen hängt nicht alleine von den verfügbaren Rezeptorwerten ab. Es können ebenfalls die Menge und die Art der Information, die ausgegeben werden soll, mit in die Berechnung einfließen. Somit würde der Tatsache Rechnung getragen, dass verschiedene Nachrichtenformate den Benutzer unterschiedlich ablenken (z.B. Video im Gegensatz zu Text oder auch Text-to-Speech verglichen mit von menschlichen Sprechern aufgezeichnete Audiodateien) und das Erfassen kurzer Inhalte tendenziell weniger Konzentration erfordert als die Aufnahme langer Inhalte. Eine einfache Bewertungsfunktion, welche die Beachtung der aktuellen Gesetzeslage in Deutschland sicherstellt, verhindert durch das Setzen des entsprechenden Safety Levels umfangreiche

visuelle Ausgaben. Bei der prototypischen Umsetzung erfolgt der Wechsel zwischen den Fahrzuständen noch manuell, um verschiedene Testfälle simulieren zu können.

Der Application Safety Manager stellt zwei Anwendungsfälle zur Verfügung: das Abfragen des aktuellen Safety Levels und das Eingeben von Umwelt- und Fahrzeuginformationen. Der Anwendungsfall „Safety Level abfragen" wird durch den Basisdienst Ausgabemanager ausgelöst (Teil der Interfacesteuerung), der die oberste Nachricht in der Ausgabe Queue wählt und den aktuellen Safety Level beim Application Safety Manager abfragt. Der Application Safety Manager gibt den aktuellen Wert des Safety Levels zurück. Die Ausgabeverwaltung gibt die betreffende Nachricht entsprechend des Safety Levels aus. Da die Nachrichten zwischen Application Safety Manager und Informationsdistributor lediglich aus einem (zeitnah zu übertragenden) Safety Level bestehen, sollte der Nachrichtentext so kurz wie möglich gehalten werden. Deshalb werden lediglich das Safety Level sowie ein Timestamp per XML an den Informationsdistributor versandt.

Der Anwendungsfall „Umwelt-/Fahrzeuginformation eingeben" wird von einem Basisdienst ausgelöst, der eine Sensoreinheit anbindet. Eine Sensoreinheit gibt nach einer Änderung ihren aktuellen Messwert die Umwelt (z.B. Regenmenge, Temperatur) oder das Fahrzeug (z.B. Reifendruck, Lane Departure Warning) betreffend an den Application Safety Manager weiter. Der Application Safety Manager berechnet aus allen vorliegenden Informationen dann das aktuelle Safety Level. Dazu stellt der Application Safety Manager den verschiedenen Fahrzustandsrezeptoren Schnittstellen bereit, über welche die Rezeptoren Informationen, die die Fahrsituation betreffen, an den Application Safety Manager weitergeben. Diese Schnittstellen sind auf den jeweiligen Rezeptor-Typ abzustimmen und zu erstellen.

Mögliche Typen für Rezeptoren können Geschwindigkeitsmesser, ein Navigationssystem, Regensensoren, Sensoren für den Reifendruck, Lane Departure Warning, Toter Winkel Sensor, das Erkennen von Überholmanövern oder ein Drucksensor für Gas/Bremspedal sein. Konzeptionell soll die Erweiterung um beliebige Sensoren für Informationen aus der Umwelt und/oder aus dem Fahrzeug (Ultraschall, Radar, Kamera etc.) möglich sein. Selbst die Integration von Sensordaten, die den Zustand des

Fahrers selbst betreffen, sind denkbar. So könnte durch Messung der Lidschlagfrequenz auf Müdigkeit, durch Messung des Pulses oder der Hautfeuchtigkeit auf Stress oder Anstrengung geschlossen werden.

2. *Ausgabemanager/Outputmanager*

Der Outputmanager dient als Schnittstelle zwischen den im System befindlichen Diensten und der Ausgabe zum Benutzer. Seine Aufgabe ist die Verwaltung der Ausgabegeräte, die einen direkten Einfluss auf die Reihenfolge der Nachrichten unterschiedlicher Dienste ermöglicht. Dazu zählt ebenfalls die Nutzung des Application Safety Managers, durch den die Fahrsicherheit der Dienste aufrechterhalten wird.

Dienste geben ihre erzeugten oder beispielsweise über UMTS, GSM oder DAB empfangenen Nachrichten über den Outputmanager an den Nutzer weiter. Da jedoch nicht alle Daten zu jeder Zeit über jeden Kanal an den Nutzer weitergegeben werden sollten, bedarf es einer Kontrollinstanz. Diese wird durch den Application Safety Manager realisiert, der ständig über die aktuelle Situation informiert ist und dem Dienst die entsprechenden Ausgabemöglichkeiten gegebenenfalls zur Verfügung stellt. Entsprechend der Rückmeldung des Application Safety Managers wird das oder die Ausgabegerät(e) gewählt, um dort die Informationen wiederzugeben. Abschließend wird eine Erfolgs- oder Fehlermeldung an den Dienst geschickt. Der Sollablauf der Informationsweitergabe beginnt mit dem Empfang der auszugebenden Daten vom auslösenden Dienst. Der Outputmanager erfragt darauf den aktuellen Safety Level beim Application Safety Manager, wählt entsprechend das oder die Ausgabegerät(e) und gibt anschließend dort die Daten aus. Erfolg oder Misserfolg werden an den auslösenden Dienst gemeldet.

Die von den Diensten an den Ausgabemanager gesandten Nachrichten müssen alle wesentlichen Informationen enthalten, die zur Klassifizierung der Nachricht und zur korrekten Eingliederung in eine Ausgabehierarchie dienen. Um die Klassifizierung und Eingliederung in die Hierarchie konsistent gewährleisten zu können, werden Diensten spezielle Kennungen zugewiesen (hier: zentral vom Service Provider), Hierarchieinformationen werden in einer separaten Look up-Tabelle vorgehalten. Die Verwendung einer separaten Tabelle für die Hinterlegung der Hierarchieinfor-

mationen hat den Vorteil, dass bei einer Änderung, z.B. durch Hinzukommen oder Wegfall eines Dienstes, nicht alle übrigen Dienste angepasst werden müssen. Damit wird zum einen der Aufwand für die Eingliederung neuer Dienste reduziert, zum anderen bleibt die Information immer konsistent.

Damit jeder Kommunikationspartner, also der ausgebende Dienst, der Ausgabemanager und sämtliche Basisdienste der Ausgabe, eine Nachricht eindeutig referenzieren können, vergibt der Dienst eine dienstintern eindeutige Kennung an jede gesandte Nachricht. Eine intern eindeutige Kennung macht einen weiteren Dienst (Naming) überflüssig und unterstützt somit den streng modularen Aufbau der Architektur sowie die Möglichkeit, Dienste einer sehr heterogenen Gruppe von Anbietern aufzunehmen. Diese Art von Kennung ist dabei ausreichend, um Nachrichten systemweit eindeutig zu referenzieren, da die (eindeutige) Dienstekennung als Namensraum betrachtet werden kann. Dieses Prinzip wird z.B. im Internet erfolgreich eingesetzt. Die Länderkennungen entsprechen hier den global eindeutigen Dienstkennungen, der Domainname entspricht der Nachrichtenkennung.

Zu den weiteren Bestandteilen des Nachrichtenformates gehört ein Feld, das die Gültigkeitsdauer (Time-To-Live, TTL) der Nachricht angibt. Damit ist es dem Dienst möglich, indirekt auf die Ausgabe seiner Nachricht Einfluss zu nehmen. Die Time-to-Live einer Nachricht gibt an, nach welcher Zeitdauer (in Millisekunden) die Nachricht nicht mehr ausgegeben werden soll. Dies ist vor allem bei Diensten mit zeitkritischen Informationen (Navigation, Hinweis auf POI, …) extrem relevant, da hier eine verspätete Information mitunter zu einer Gefährdung des Benutzers führen kann. Die Ausgabe der Nachricht „jetzt rechts Abbiegen“ könnte aufgrund einer höher priorisierten Ausgabe (z.B. „Temperatur des Motors kritisch“) verzögert werden und just dann erfolgen, wenn der Fahrer an einer Einbahnstraße vorbei fährt.

Zwei weitere Felder des Nachrichtenformates nehmen Einfluss auf die Ausgabe: Zum einen wird der auszugebende Inhalt übergeben, zum anderen werden das gewünschte Ausgabemedium oder die gewünschten Ausgabemedien angegeben. Der Inhalt kann dabei direkt angegeben werden (z.B. Text) oder als Referenz übergeben werden (z.B. XLink auf Bilder, Videos etc.). Die gewünschten Ausgabemedien wer-

den als Parameter mit übergeben, die Konsistenzprüfung erfolgt dabei durch den ausgebenden Dienst selbst. Diese dezentrale Konsistenzprüfung verteilt die Last auf alle ausgebenden Dienste und entlastet damit den zeitkritisch operierenden Ausgabemanager zusätzlich.

a) Sprachsynthese/Speechengine

Die Schnittstelle vom Ausgabemanager zur Sprachausgabe unterliegt nicht denselben Einschränkungen, die bei der grafischen Ausgabe von Bedeutung sind. Es gibt zwischen den Herstellern keine Unterschiede, wie viel Information auf einmal dargestellt werden kann. Der einzige Unterschied liegt in der verwendeten Stimme und prosodischen Regeln, welche Teil der Corporate Identity sind.

Für die Sprachausgabe (Text to Speech) des Prototypen wurde auf SVOX zurückgegriffen. Die Software wird bereits im Automotive Sektor eingesetzt und hat die Eigenschaft, dass sie Sprachmechanismen an sich von der konkreten Sprache losgelöst betrachtet. Die Grundfunktion der Sprachsynthese bleibt gleich, selbst wenn der eingegebene Text in einer anderen Sprache verfasst wurde. Bislang wird von SVOX jedoch der SSML (Sprach Synthese Markup Language) Standard nicht unterstützt. Die Verwendung solcher Standards ist zukünftig jedoch für die herstellerübergreifende Nutzbarkeit von Diensten unabdingbar. Für die Zukunft ist diese Funktion bei SVOX angekündigt. Bis SSML unterstützt wird, muss durch geeignete Konverterprogramme standardisierter SSML Text in das proprietäre SVOX Format überführt werden.

Obwohl die Informationsausgabe mittels Sprachsynthese nicht dasselbe Ablenkungspotenzial wie Textausgabe oder gar das Abspielen eines Videos darstellt, kann eine plötzlich ertönende Stimme in einer kritischen Fahrsituation zu einem stark erhöhten Unfallrisiko führen. Aus diesem Grund unterliegen auch Sprachausgaben denselben Regelungen des Application Safety Managers und Ausgabemanagers wie grafische Ausgaben.

b) Displayausgabe/Display Renderer

Das Layout und die Funktionalität der Displayausgabe wurden in Java mit Swing nachgebildet. Da das Seitenverhältnis des Display im Fahrzeug von dem eines Standard-PCs abweicht, wird die Ausgabe horizontal gestreckt, damit diese bei der vertikalen Streckung nach der Ausgabe im Fahrzeug „normal" erscheint. Zur Ausgabe auf dem Fahrzeugdisplay wird der Videoeingang des Fahrzeugs verwendet. Das Design der Mensch-Maschine-Schnittstelle wurde dabei stark an das bestehende Konzept des Automobilherstellers angelehnt. Zum einen hat dies den Vorteil, dass bestehende Erkenntnisse und Best Practices zur Gestaltung der Mensch-Maschine-Schnittstelle vom Automobilhersteller übernommen werden konnten. Zum anderen fügen sich neue Dienste dadurch nahtlos in das bestehende Bedienkonzept ein; ein Neu- oder Umlernen durch den Benutzer ist nicht nötig.

Für die Displayanzeige ist das aktuelle Safety Level relevant. Der Fahrer soll nicht durch die Anzeige von zu viel Text vom Geschehen im Straßenverkehr abgelenkt werden. Wird eine Ausgabe von einem Dienst angefordert, wird geprüft, ob das Safety Level eine Displayausgabe erlaubt. Der Sollablauf einer Displayausgabe beginnt mit dem Empfang der auszugebenden Daten. Anschließend wird der aktuelle Safety Level beim Application Safety Manager abgefragt. Stimmt das Level, so wird die Information auf dem Display angezeigt oder ein Fehler zurückgegeben.

Das für die MACS Diensteplattform anfänglich verwendete MMI Display, in der ersten Version des Prototypen nach der ersten Iteration (vgl. Abbildung 4) zunächst simuliert, liegt dabei in einem breiten Spektrum denkbarer Medien, die von höchstauflösenden TFT-Monitoren mit weit über 2 Millionen Bildpunkten bis hin zu Displays von Mobiltelefonen mit einigen zehntausend Bildpunkten reichen. Neben den technischen Aspekten spielen auch Anforderungen aus dem Marketing eine wesentliche Rolle, da dem Benutzer eine konsistente Form der Corporate Identity präsentiert werden sollte.

c) Weitere Ausgabedienste

Durch die offene Schnittstelle für Ausgabedienste können die Basisdienste um beliebige weitere Ausgabegeräte erweitert werden, indem für diese jeweils ein Dienst entwickelt und eingebunden wird, der dem Outputmanager über die offene Schnittstelle den Zugriff auf das Gerät, beispielsweise über Gerätetreiber oder weitere Software, ermöglicht. Gerätespezifische Logik wird dabei im jeweiligen Dienst gekapselt und die Funktionalität des Gerätes wird auf die Schnittstelle abgebildet. So können zum Beispiel ein Head Up Display, das situationsabhängig auf die Windschutzscheibe Informationen projiziert, oder ein Forcefeedback-Lenkrad oder -Pedale durch solche Wrapperdienste mit geringem Aufwand unter Aufrechterhaltung der Fahrsicherheit eingebunden werden.

3. Eingabemanager/Inputmanager

Der Inputmanager dient als Schnittstelle zwischen den im System befindlichen Diensten und der Eingabe des Benutzers. Seine Aufgabe ist die Verwaltung der Eingabegeräte sowie das Aufrechterhalten der Interaktionssicherheit. Im Gegensatz zum Ausgabemanager greift der Application Safety Manager beim Inputmanager nicht ein. Der Benutzer sollte sein Verhalten selbst besser auf das Gefahrenpotenzial der Situation abstimmen können und wäre durch Veränderungen des Systems abgelenkt, wenn dieses plötzlich ein anderes Verhalten zeigen würde, wenn z.B. das haptische Interface auf einmal „nicht mehr funktionieren“ würde.

a) Spracheingabe/Speechlistener

Zur automatischen Spracherkennung (Automatic Speech Recognition, ASR) wird die Software Vocon 3200 von Scansoft verwendet, die speziell für die sichere Bedienung von Navigationssystemen, Telematikanwendungen und anderen Geräten im Automobil entwickelt wurde. Großer Wert wurde bei der Entwicklung auf die Verbesserung von Mensch-Maschine-Schnittstellen mit dem Schwerpunkt auf Voice Activated Dialing (VAD) sowie Voice Destination Entry (VDE) gelegt. Diese Techniken bewährten sich ebenfalls in der Sprachbedienung von MP3-Playern und anderer Unterhaltungselektronik. Durch die Lauffähigkeit auf integrierten Geräten und die hohe Erken-

nungsraten, erreicht durch situationsabhängige grammatik- oder diktatbasierte Spracherkennung, ist diese ASR-Software ideal für eine Diensteplattform.

Die Informationseingabe mittels Spracherkennung führt zu geringerem Ablenkungspotenzial als die durch haptische Eingabe, da diese häufig zusätzlich einen Blick auf ein Display oder andere Anzeigen erfordert. Der Sollablauf wird durch das Drücken des PTT-Tasters (Push-tTo-Talk) am Lenkrad ausgelöst; der Fahrer muss die Hand nicht vom Lenkrad nehmen. Die Spracheingabe wird aufgezeichnet und anschliessend an die Spracherkennungssoftware übergeben; der zurückgegebene Text wird auf Befehle und deren Parameter abgebildet und an den Eingabemanager übergeben. Das erfolgreiche Erkennen und Ausführen eines Befehls bzw. der Fehler wird durch den Ausgabemanager über Sprachsynthese oder ein Tonsignal quittiert.

b) Haptische Eingabe/MMI Listener

Zur haptischen Eingabe wurde ein Dienst zur Anbindung des fahrzeuginternen Frontcontrollers sowie eine Platine zur Umwandlung der Signale in eine für PCs übliche serielle RS232-Schnittstelle einwickelt. Um das proprietäre Protokoll des Frontcontrollers empfangen und senden zu können, war hierfür die Entwicklung eines eigenen Parsers, sowie des dazugehörigen Writers und eines diskreten Zustandsautomaten (DEA) nötig. Diese enthalten Algorithmen zur Fehlererkennung in der Übertragung, so dass Nachrichten erneut gesendet werden, wenn der Empfänger bei der Überprüfung der Parität einen Fehler meldet. Die empfangenen Nachrichten werden auf die Befehle und ihre Parameter abgebildet und an den Inputmanager übergeben.

c) Weitere Ausgabedienste

Analog zu den Ausgabediensten können die Basisdienste durch die offene Schnittstelle für Eingabedienste um beliebige weitere Eingabegeräte erweitert werden, indem für diese jeweils ein Dienst entwickelt und eingebunden wird, der dem Inputmanager über die offene Schnittstelle den Zugriff auf das Gerät beispielsweise über Gerätetreiber oder weitere Software ermöglicht. Gerätespezifische Logik wird dabei im jeweiligen Dienst gekapselt und die Funktionalität des Gerätes auf die Schnittstel-

le abgebildet. So können zum Beispiel Maus, Tastatur oder ein Joystick durch derartige Wrapperdienste mit geringem Aufwand eingebunden werden.

4. Daten- und Persistenz Manager

Wesentlicher Bestandteil einer Diensteplattform für mobile Dienste oder Telematikanwendungen, ist die persistente Datenhaltung sowie der (modularisierte) Zugriff auf Daten über Telekommunikationsmedien. Analog zu Ein- und Ausgabemanager, die die unterschiedlichen Benutzerschnittstellen vereinheitlichen, bietet der Daten- und Persistenzmanager die Möglichkeit, unterschiedliche Übertragungswege in der Diensteplattform zu integrieren; er macht diese damit auch für zukünftige Dienste nutzbar.

Die grundlegende Idee hinter dem Daten- und Persistenzmanager ist, dass Dienste nicht mehr spezifische Hardwarekomponenten im Fahrzeug, z.B. das Radio oder ein Mobiltelefon, direkt ansprechen, sondern vielmehr über verschiedene URIs (Uniform Ressource Identifier)[70] auf Daten zugreifen. Aufgabe des Daten- und Persistenzmanagers ist es, auf eine URI Anfrage zu reagieren, indem aus dem Schema-Teil der URI die passende Hardwarekomponente zur Bearbeitung der Anfrage per Lookup ermittelt wird. Als Beispiel für die mögliche Umsetzung im Rahmen einer Diensteplattform im Automobilbereich ist die von Vaha-Sipila beschriebene Ansteuerung von Telefonen für Wählverbindungen.[71]

a) Universeller Handyadapter

Der Dienst „Universeller Handyadapter“ steht im Zusammenhang mit der Nutzung von Mobiltelefonen oder weiterführenden Netzwerkverbindungen. Andere Dienste können diesen Dienst nutzen, um beispielsweise die Verfügbarkeit von Netzwerken abprüfen zu können, um ggf. den Serviceumfang einzuschränken oder den Benutzer darauf hinweisen zu können, dass der Dienst derzeit nicht verfügbar ist.

Der universelle Handyadapter ist, wie sein Name schon sagt, der Basisdienst, der die Verbindung der Dienste auf der MACS MyServices Plattform mit dem GSM Netz be-

---

[70] Vgl. Hansen et al. (2006); Daigle et al. (2002).
[71] Vgl. Vaha-Sipila (2000).

reitstellt. Der Fokus liegt dabei auf der hardwarenäheren Seite, es können also die Besitzerinformationen des Handy (SIM Access Profile) und die Verbindung mit dem GSM Netz (Netzstatus) genutzt bzw. überprüft werden. Die Funktionen des universellen Handyadapters werden durch jene aus dem Basisdienst „Connectivity" vervollständigt, der das Übertragungsmedium (z.B. DAB, GSM etc.) komplett kapselt.

Schnittstellen vom Dienst zum universellen Handyadapter sind einfache Java Methodenaufrufe, welche als Parameter die nötigen Informationen an den Handyadapter übergeben. Für die einfachen Fälle der Verfügbarkeitsprüfungen von Mobilfunkanbindung (bestehend aus den Bedingungen „Verfügbarkeit des Mobiltelefons" sowie „in einer Mobilfunkzelle angemeldet") sind z.B. keine Übergabeparameter zwingend erforderlich. Für den Zugriff auf die Informationen der SIM Speicherzellen über das SIM Access Profile (von der Bluetooth Special Interest Group als Möglichkeit vorgeschlagen) sind einige Parameter nötig, welche in der entsprechenden Dokumentation präzise erläutert werden.

Der allgemeine Ablauf des SIM Zugriffs ist wie folgt: um den Transfer zu ermöglichen, müssen neben den Kriterien, dass eine SIM Karte und Bluetooth verfügbar sind, auch das Personal Area Network (PAN) oder das LAN Access Profile vom Gerät unterstützt werden. Beim Datentransfer übernimmt das Serviceportal auf der Onboard Telematics Unit die Rolle des SIM Access Client und initiiert die Verbindung zum SIM Access Server, dem Mobiltelefon. Nachdem der Benutzer (z.B. durch Eingabe seiner PIN) die Verbindung autorisiert hat, kann der Client (Serviceportal) die SIM Karte im Server komplett auslesen und somit die Identität übernehmen. Da die SIM Karten als einzigartige Benutzer IDs gegenüber den Mobilfunkbetreibern dienen, bietet sich so die Möglichkeit die Identität eines Benutzers anhand der Daten, die von seiner SIM Karte ausgelesen wurden, festzustellen.

Nach dem aktuellen Stand ist es eher unwahrscheinlich, dass in Fahrzeugen im Premiumsegment keine eigene Mobilfunkeinheit verbaut werden wird. Ein universeller Handyadapter ist in diesem Fall nicht zwingend notwendig, da jedoch die bereitgestellte Funktionalität (besonders die Verfügbarkeitsprüfung der Mobilfunkanbindung)

immer noch benötigt wird, kann die Implementierung einer universellen Lösung als guter modularer Aufbau einer Diensteplattform gesehen werden.

b) Universeller Rundfunkadapter

Der universelle Rundfunkadapter stellt Dienste vergleichbar dem universellen Handyadapter zur Verfügung; allerdings sind diese auf das Medium Rundfunk abgestimmt: Aufgrund des fehlenden Rückkanals vom Fahrzeug zum Diensteanbieter können Daten nicht direkt angefordert werden und nicht (mehr) verfügbare Daten können nur über Timeouts erkannt werden, nicht über Fehlermeldungen vom System.

Der allgemeine Ablauf zum Abrufen von Informationen über den universellen Rundfunkadapter beginnt mit der Übergabe von Verbindungsparametern, z.B. Radiofrequenz oder Ensemble, sowie Pfad- und Dateiinformationen. Entsprechend der Frequenz oder des Ensemble stellt der Tuner des Radioempfängers die entsprechende Station ein und speichert die Daten im Cache zwischen. Wurde die angefragte Datei empfangen, gibt der universelle Rundfunkadapter diese zurück an den aufrufenden Dienst. Optional empfängt der Rundfunkadapter weiter den Datenstrom der Sendeanstalt und gibt bei jedem Update der angeforderten Datei diese zurück bis ein neuer Aufruf des Dienstes erfolgt, andernfalls wird der Tuner wieder deaktiviert.

Im Gegensatz zu direkten Verbindungen, wie z.B. über das Mobiltelefon, ist es bei Rundfunkdaten weder möglich Daten direkt anzufordern, noch zu erkennen, ob bestimmte Daten überhaupt verfügbar sind. Dementsprechend müssen Timeouts sinnvoll gewählt werden, um nicht zu lange auf die Übertragung einer Datei zu warten, die nicht existiert, aber dennoch lange genug um einige Sendezyklen abzudecken. Im Idealfall erhält man durch enge Kooperation mit dem Betreiber des Rundfunknetzes detaillierte Informationen über die Größe (und damit Ausstrahlungsdauer) des betreffenden „Karussells“.

c) *Weitere Dienste*

Die Daten-/Persistenzdienste können ebenfalls durch die offene Schnittstelle um beliebige weitere Geräte erweitert werden, indem jeweils ein neuer Dienst entwickelt und eingebunden wird, der dem Daten-/Persistenzmanager über die offene Schnittstelle den Zugriff auf das Gerät beispielsweise über Gerätetreiber oder weitere Software ermöglicht. Gerätespezifische Logik wird dabei im jeweiligen Dienst gekapselt und die Funktionalität des Gerätes wird auf die Schnittstelle abgebildet. Von besonderer Bedeutung sind Vorgehensweisen und Methoden wie sie bereits von Daigle beschrieben werden: über Uniform Ressource Identifier (URI) werden Ressourcen sowie Zugriffsschemata definiert.[72]

5. *Software Manager*

Der Software Manager bietet Dienste zum Aufspielen verschiedener Programme an. Neben der Installation weiterer Dienste bieten die einzelnen Dienste in diesem Paket auch die Möglichkeit, Servicebundles zu installieren sowie das komplette System zu aktualisieren oder zu erweitern. Das ermöglicht, Updates der Diensteplattform einzustellen, aber auch neue Dienste für ein Fahrzeug zu kaufen und dort zu installieren.

Die Softwareinstallation auf der MACS Diensteplattform erfolgt analog zur Installation eines neuen Bundle im OSGi Framework. Die Installation läuft nach einem fixen Schema ab, wobei sich die drei möglichen Ausprägungen lediglich in der Art und Weise unterscheiden, in der die Software zur Verfügung gestellt wird. Die erste Möglichkeit ist der Broadcast der Software über DAB, sie eignet sich vor allem für kleine Anwendungen (d.h. die Downloadzeit ist gering), welche die (für alle Anwender gleiche) Diensteplattform betreffen. Eine weitere Möglichkeit, Software in das Fahrzeug zu übertragen, ist der Download per GSM/GPRS/UMTS. Vorteil dieser Methode ist, dass die Installation an jedem beliebigen Ort (mit Mobilfunk-Netzabdeckung) und sogar automatisiert während der Fahrt erfolgen kann. Sicherheitsrelevante Software sollte immer vor Ort in einer Vertragswerkstatt von einem Techniker installiert wer-

---

[72] Vgl. Daigle et al. (2002).

den. Updates für bereits auf dem System vorhandene Daten können und sollten Teil des Servicezyklus werden.

Es werden bei der Softwareinstallation keine gesonderten Schnittstellen zwischen Dienst und Benutzer oder zwischen Dienst und Hardware benötigt. Vielmehr ist es ausreichend, die Schnittstellen des Connectivity Paketes sowie des OSGi Frameworks zu verwenden.

3.1.5 Test und Evaluation

Bei den auf die einzelnen Phasen abgestimmten Tests während der Entwicklung der MACS Diensteplattform wird auf die in Kapitel 2.4.4 beschrieben Methoden zurückgegriffen. Dabei wird für jede einzelne Iteration des Spiralmodells (vgl. Abbildung 2) ein kompletter Satz Tests durchgeführt, unterteilt in die Phasen der Implementierung, der Analyse und des Systementwurfes sowie der Anforderungserhebung.

Bereits während der Implementierung wurden die einzelnen Programmeinheiten regelmäßig durch Unit Tests, d.h. in Isolation von anderen im Zusammenhang eines Programms benötigten, mitwirkenden Programmeinheiten, auf ihre Korrektheit geprüft, um die Erfüllung funktionaler Anforderungen zu gewährleisten.[73] Die Unit Tests wurden dabei gemäß der Programmiertechnik der testgetriebenen Entwicklung[74], bereits vor der Implementierung der jeweiligen Programmeinheiten entwickelt, so dass diese inkrementell gemäß den Anforderungen verändert werden konnten.

Zur automatisierten Durchführung der Unit Tests wurde das von Kent Beck und Erich Gamma entwickelte JUnit-Framework eingesetzt. Die Unit Tests der Programmeinheiten enthalten mehrere isoliert voneinander durchführbare Testfälle, die nach jedem erfolgreichen Kompilieren in einem Testlauf ausgeführt werden. Dazu wurden im Build-Tool Ant der Apache Software Foundation unterschiedliche Tasks in verschiedene Targets zusammengefasst. Ziel ist die automatische Bereinigung der Entwicklungsumgebung vor dem Kompillieren, das automatisierte Durchführen von Junit

---

[73] Vgl. Brügge/Dutoit (2000), S. 334f.
[74] Vgl. u.a. Beck (2003); Jeffries et al. (2001), S. 117 ff; Fowler (2006).

Tests nach der Kompillierung sowie das Generieren der OSGi Bundles und der Programmdokumentation ohne manuellen Eingriff des Entwicklers. Das Ergebnis der Unit Tests sind Ausgaben zu Fehlerfällen in den einzelnen Objekten und deren Klassen. Auf Basis dieser Information lassen sich einzelne Komponenten isoliert fehlerfrei implementieren.

Im Rahmen der Anforderungsanalyse und des Systementwurfes steht nicht mehr die konkrete Umsetzung auf dem Prüfstand; Unit Tests sind damit nicht ausreichend um die erarbeiteten Anforderungen zu überprüfen. Aus diesem Grund wird mit System- und Modultests überprüft, ob das implementierte System den erhobenen Anforderungen entspricht. Angelehnt an Brügge und Dutoit kommt eine Black-Box-Technik zum Einsatz, bei der zu jeder fertigen Komponente, also jedem OSGi Bundle, des Systems ein eigenes Testbundle erstellt wird, das im Vorfeld festgelegte Eingaben für die vom Bundle erbrachten Dienste sukzessive überprüft.[75] Dieser Prozess kann nicht mehr für alle Fälle automatisiert erfolgen; gerade die Überprüfung der Komponenten zur Benutzerintegration, also Ein- und Ausgaben, müssen manuell überprüft werden. Nach erfolgreichem Bestehen der Black-Box-Tests aller Komponenten besteht ein sowohl korrekt implementiertes als auch korrekt entworfenes System; es bleibt die Überprüfung der Korrektheit der Anforderungserhebung.

Die Evaluation der Nützlichkeit der MACS Diensteplattform, also der Korrektheit der erhobenen Anforderungen, erfolgte implizit durch die Implementierung von MACS MyNews anstelle expliziter Tests. MACS MyNews baut auf der MACS Diensteplattform auf. Der Entwicklungsprozess wird im folgenden Kapitel eingehend behandelt.

3.2 MACS MyNews

Im folgenden Kapitel wird die Anwendung der Ebenen „Technologie“ sowie „Prototyp und Plattform“ des MACS Design Framework (siehe Abbildung 1) auf einen exemplarischen Dienst nach dem entsprechenden Vorgehensmodell beschrieben. Dabei werden jeweils insbesondere die Erkenntnisse der einzelnen Stufe auf dem Weg zu

---

[75] Vgl. Brügge/Dutoit (2000), S. 359.

einem ausrollbaren mobilen Dienst für das Automobil herausgearbeitet. Erster Schritt dazu ist die Auswahl eines passenden Dienstes, der für eine Umsetzung Erfolg versprechend scheint. Darauf aufbauend werden, auch anhand der Annahmen der Diensteplattform, die Anforderungen analysiert und das System entworfen. Im folgenden Schritt wird der Systementwurf zu einem test- und evaluierbaren Prototypen implementiert.

3.2.1 Auswahlprozess MACS MyNews

Als Basis der Auswahl dient ein von Mercer Management Consulting erhobenes Spektrum mobiler Dienstleistungen mit besonderen Fokussierung auf denkbare personenbezogene Dienste (vhl. Abbildung 15).[76] Aus den Diensten des Forschungsspektrums wurde ein personalisierbarer Nachrichtendienst, MACS MyNews, aufgrund des Kundennutzen, der Bepreisbarkeit, des Innovationsgrades sowie der technischen Realisierbarkeit eines Prototypen als aussichtsreichste Idee bewertet und umgesetzt.

Abbildung 15 Spektrum mobiler Dienstleistungen[77]

---

[76] Vgl. Ehmer (2002), S. 469.
[77] Vgl. Ebenda.

3.2.2 Anforderungserhebung

Das Diensteszenario: MACS MyNews ist ein personalisierbarer, interaktiver Nachrichtendienst, der es den Nutzern erlaubt sich als Redakteur und Endnutzer ihrer eigenen Nachrichtensendung zu sehen. MACS MyNews stellt jederzeit aktuelle Informationen bereit. Der Fahrer kann als „Chefredakteur" seine Interessensgebiete nach Themen sortiert wählen, die Reihenfolge der Nachrichten bestimmen, Gewichtungen der Kategorien vornehmen etc. Die Nachrichtensendung selbst beginnt „auf Knopfdruck" und kann jederzeit pausiert werden, z.B. falls der Fahrer tanken muss. Es ist auch möglich mit MACS MyNews zu interagieren, also zwischen Nachrichten vor und zurück zu springen, bzw. sich eine Nachricht nochmals vorlesen zu lassen, falls man ein Detail der Nachricht verpasst hat.

Es gibt bei MACS MyNews nur einen realen Benutzer. Da die Nutzung des Dienstes jedoch von der Situation, in der sich der Benutzer befindet, abhängig ist, wird beim Benutzer im Folgenden zwischen den situationsbedingten Rollen „Fahrer" und „Nutzer" unterschieden (vgl. Abbildung 16). Der Benutzer wird vom Nutzer zum Fahrer, sobald sich das Fahrzeug mit mehr als 5 km/h bewegt. Umgekehrt wird der Fahrer zum Nutzer, sobald das Fahrzeug langsamer als 5 km/h wird. Der Nutzer stellt also eine Spezialform des Fahrers dar, er darf auch Funktionalitäten des Newsdienstes nutzen, deren Benutzung in der Fahrsituation zu gefährlich oder gesetzlich verboten ist. Die jeweiligen Gründe, weshalb der Benutzer in seiner Rolle als Fahrer eine Funktionalität nicht nutzen kann, werden gegebenenfalls bei dem entsprechenden Anwendungsfall im folgenden Kapitel erläutert.

Fahrer Nutzer

Abbildung 16 Generalisierung der Rolle „Fahrer"

Sämtliche Anwendungsfälle des MACS Newsdienstes sind in Abbildung 17 als Übersicht dargestellt. Abbildung 18 zeigt die Anwendungsfälle „Profil erstellen" und „Profil

anpassen“ nochmals genauer. Detaillierte Beschreibungen der einzelnen Anwendungsszenarien folgen in diesem Kapitel.

Abbildung 17 Übersicht der MACS MyNews Anwendungsfälle

1. *Anwendungsfall: Einstieg in den Dienst*

Akteure: Fahrer, Nutzer

Vorbedingungen: MACS MyNews inaktiv

Nachbedingungen: MACS MyNews aktiv

Beschreibung: Der Fahrer oder Nutzer des Dienstes startet per Knopfdruck oder per Sprachbefehl den MACS MyNews Dienst. Ist ein persönliches Profil vorhanden, beginnt sofort die Präsentation der gewählten Nachrichten. Ist kein Profil verfügbar, beginnt eine allgemeine Präsentation der Nachrichten. Dem Benutzer wird angesagt, ob mit oder ohne Profil gestartet wird.

2. Anwendungsfall: Direktauswahl von Nachrichten

Akteure: Fahrer, Nutzer

Vorbedingung: MACS MyNews ist aktiv, kein Profil ist verfügbar

Beschreibung: Ist im Fahrzeug kein Profil verfügbar, so besteht die Möglichkeit, aus den kompletten Nachrichten gezielt einzelne Nachrichteninhalte abzurufen. Dazu werden zunächst von MACS MyNews die verfügbaren Kategorien angezeigt und/oder vorgelesen. Der Fahrer/Nutzer kann dabei jederzeit eine Kategorie auswählen oder zwischen Kategorien wechseln (Vorlesen überspringen bzw. nochmals vorlesen lassen). Nach demselben Schema werden bei Auswahl einer Kategorie die entsprechenden Unterkategorien ausgewählt. Ist eine Nachricht ausgewählt, wird dem Fahrer/Nutzer der Inhalt präsentiert. Falls ein Mobiltelefon verfügbar ist, werden dem Benutzer verfügbare Bezahlinhalte zum Herunterladen und Abspielen angeboten.

Sonderfälle: Es gilt die Einschränkung, dass dem Fahrer nur in der Fahrsituation verwendbare Bezahlinhalte zur Kenntnis gebracht werden. Konkret bedeutet dies, dass dem Fahrer lediglich weitere Artikel, welche vom Newsdienst vorgelesen werden, und Audiodateien angeboten werden. Für den Nutzer gilt diese Einschränkung nicht; er bekommt alle Bezahlinhalte angeboten, also neben Texten und Audidateien auch Videos und Bilder.

3. Anwendungsfall: Nachrichten vorlesen lassen

Akteure: Fahrer, Nutzer

Vorbedingung: MACS MyNews aktiv, Nachrichteninhalte verfügbar, für Bezahlinhalte: Mobiltelefon verfügbar

Beschreibung: Da der Fahrer aus Sicherheitsgründen das Display zum Lesen von Nachrichten nicht benutzen darf, stellt dies die einzige Möglichkeit für ihn dar, die Nachrichteninhalte vermittelt zu bekommen. Der Fahrer/Nutzer muss während des Vorlesens der Nachricht jederzeit die Möglichkeit haben, zur nächsten oder vorherigen Nachricht zu springen (falls verfügbar), das Vorlesen zu pausieren und wieder aufzunehmen sowie sich die aktuelle Nachricht nochmals von vorne vorlesen zu lassen. Auch der Abbruch des Vorlesens muss für den Fahrer/Nutzer möglich sein. Dem Fahrer/Nutzer steht ebenfalls die Möglichkeit offen, die aktuelle Nachricht

zur weiteren Bearbeitung an seine Mailadresse senden zu lassen, sofern sein Mobiltelefon verfügbar ist und der Newsdienst entsprechend konfiguriert wurde.

Sonderfälle:

- Sind Bezahlinhalte mit einer vorgelesenen Nachricht verknüpft, wird der Benutzer darauf hingewiesen, falls sein Mobiltelefon verfügbar ist und die Inhalte in der aktuellen Rolle genutzt werden können. Der Fahrer erhält also nur Textnachrichten (die vorgelesen werden) und Audiodateien, der Nutzer bekommt auch den Hinweis auf Videos und Bilder.
- Die Standardansage des Dienstes bei Bezahlinhalten besteht aus dem Typ des Bezahlinhaltes und dem Titel der beim Inhalt hinterlegt wurde (Beispiel: „Audio Mehrinhalt verfügbar: ‚Interview mit Helmut Kohl'").
- Nach der Standardansage wird eine kurze Zeit auf eine Benutzereingabe gewartet. Erfolgt keine Benutzereingabe, wird für Flatrate-Nutzer der Bezahlinhalt herunter geladen und abgespielt und/oder angezeigt, für alle übrigen Benutzer nicht.

4. Anwendungsfall: Nachrichten anzeigen lassen

Akteur: Nutzer

Vorbedingung: MACS MyNews aktiv, Nachrichteninhalte verfügbar, für Bezahlinhalte: Mobiltelefon verfügbar, Fahrzeug langsamer als 5km/h

Beschreibung: Für den Nutzer gibt es die Möglichkeit, sich die einzelnen Artikel im Display anzeigen zu lassen. Da der Fahrer aus Sicherheitsgründen das Display zum Lesen von Nachrichten nicht benutzen darf, ist diese Möglichkeit für ihn nicht verfügbar. Der Nutzer muss während des Lesens der Nachrichten jederzeit die Möglichkeit haben, zur nächsten oder vorherigen Nachricht zu springen (falls verfügbar). Dem Nutzer steht ebenfalls die Möglichkeit offen, die aktuelle Nachricht zur weiteren Bearbeitung an seine Mailadresse senden zu lassen, sofern sein Mobiltelefon verfügbar ist und der Newsdienst entsprechend konfiguriert wurde. Bei der Anzeige von Nachrichten werden dem Nutzer alle auswählbaren Bezahlinhalte angezeigt.

Sonderfälle:

- Sind Bezahlinhalte mit einer angezeigten Nachricht verknüpft, wird der Nutzer darauf hingewiesen, falls sein Mobiltelefon verfügbar ist.

- Der Hinweistext soll aus der Angabe des Inhaltstyps sowie dessen Titel bestehen (Beispiel „Audio: Interview mit Helmut Kohl“).
- Wären Bezahlinhalte nur durch Scrollen sichtbar, soll der Nutzer durch ein bestimmtes Zeichen beim Titel des Artikels auf die Verfügbarkeit von Zusatzinhalten hingewiesen werden.

5. Anwendungsfall: Profil erstellen

Akteur: Nutzer

Vorbedingung: MACS MyNews aktiv, Fahrzeug fährt langsamer als 5km/h

Nachbedingung: Profil gespeichert

Beschreibung: Um den MACS Newsdienst optimal nutzen zu können, muss es dem Nutzer möglich sein, ein persönliches Profil zu erstellen. Darin werden unterschiedlichste Informationen hinterlegt, unter anderem die Kategorien, aus denen der Nutzer Nachrichten erhalten möchte inklusive deren Reihenfolge und Priorisierung. Es soll dem Nutzer dabei möglich sein, einzelne Unterkategorien auszuwählen, die Reihenfolge wird aus Gründen der Übersichtlichkeit jedoch nur auf der obersten Kategorienebene festgelegt. Der Nutzer hat ebenfalls die Möglichkeit, die gewünschte Dauer seiner personalisierten Nachrichten anzugeben und zu bestimmen, wie sich die gewählte Zeit auf die einzelnen Kategorien verteilt. So soll es möglich sein, dass die gewählte Newsdauer gleich über die Kategorien verteilt wird oder aber nach der Reihenfolge den ersten Kategorien mehr Zeit zugestanden wird. Zuletzt muss der Nutzer eine PIN angeben, welche das Profil vor unberechtigtem Überschreiben schützt. Anhand eines vom Nutzer vergebenen Namens kann das zu erstellende Profil identifiziert werden.

Abbildung 18 Details zu den Anwendungsfällen „Profil erstellen“ und „Profil anpassen“

6. Anwendungsfall: Auswahl von Kategorien

Akteur: Nutzer

Übergeordneter Anwendungsfall: Profil erstellen

Beschreibung: Eine Möglichkeit, den Newsdienst zu personalisieren, ist die Auswahl der Nachrichtenkategorien, die für den Fahrer oder Nutzer von Interesse sind. Der Benutzer kann eine Kategorie aus einer Liste von Kategorien auswählen. Wählt der Nutzer eine Hauptkategorie, bei der keine Unterkategorie gewählt war, werden automatisch alle Unterkategorien aktiviert, der Nutzer kann diese Auswahl bei den einzelnen Unterkategorien danach wieder entfernen. Waren schon eine oder mehrere Unterkategorien einer Hauptkategorie aktiviert, werden bei erneuter Anwahl der Hauptkategorie nicht alle Unterkategorien selektiert. Der Nutzer hat nun die Möglichkeit, einzelne Unterkategorien auszuwählen, über den Punkt „Zurück“ gelangt

er wieder in die Ansicht der Hauptkategorien. Hat der Nutzer die Auswahl der gewünschten Kategorien abgeschlossen, gelangt er über den Menüpunkt „Zurück“ wieder in die Ebene der Profileinstellungen.

7. Anwendungsfall: Festlegen der Kategorienreihenfolge

Akteur: Nutzer

Übergeordneter Anwendungsfall: Profil erstellen

Beschreibung: Eine weitere Möglichkeit der Personalisierung ist die Festlegung der Reihenfolge der einzelnen Nachrichtenkategorien. Der Newsdienst zeigt dem Nutzer alle Hauptkategorien an, für die Unterkategorien ausgewählt wurden. Wurden die Kategorien bislang noch nicht in eine Folge gebracht (z.B. bei Ersterstellung des Profils), sind die Kategorien gemäß der Indexstruktur aufgelistet. Hat der Nutzer bereits zu einem früheren Zeitpunkt die Reihenfolge der Kategorien geändert, werden die Kategorien gemäß seiner Vorgabe aufgelistet. Hauptkategorien, die seit der letzten Festlegung der Reihenfolge neu hinzukamen, werden am Ende der Liste angestellt. Der Nutzer wählt eine Kategorie aus der Liste aus und verschiebt sie in der Liste nach unten bzw. oben. Ist die Kategorie am gewünschten Platz, bestätigt das der Nutzer, um sie zu deselektieren. Ist die Festlegung der Reihenfolge der Kategorien abgeschlossen, gelangt der Nutzer mit dem Befehl „Zurück“ wieder zu seinen Profileinstellungen, andernfalls hat er die Möglichkeit, weitere Kategorien anzuwählen und zu verschieben.

Sonderfälle: Kategorien, die den Anfang der Liste bilden und nach oben verschoben werden, kommen an das Ende der Liste; analog Kategorien, die das Ende der Liste bilden.

8. Anwendungsfall: Newsdauer eingeben

Akteur: Nutzer

Übergeordneter Anwendungsfall: Profil erstellen

Beschreibung: Die dritte Möglichkeit, den Newsdienst individuell auf eigene Bedürfnisse einzurichten, besteht in der Auswahl der gewünschten Nachrichtendauer. Der Nutzer hat dabei die Möglichkeit, einzustellen, für wie lange er Nachrichten hören möchte. Dazu werden ihm voreingestellte Zeitintervalle (z.B. 10,

20 oder 30 Minuten) vorgeschlagen, es besteht jedoch ebenfalls die Möglichkeit, die Newsdauer frei zu wählen. Der Nutzer kann mittels der „Zurück“-Option wieder zu den Profileinstellungen wechseln.
Invarianten: Die Nachrichtendauer wird als natürliche Zahl größer 0 angegeben.

9. Anwendungsfall: Gewichtung der Kategorien

Akteur: Nutzer
Übergeordneter Anwendungsfall: Profil erstellen
Beschreibung: In engem Zusammenhang mit der Priorisierung der Kategorien und der Eingabe der gewünschten Nachrichtendauer steht die Auswahl der Gewichtungsfunktion für die einzelnen Kategorien. Der Nutzer hat bei der Gewichtung der Kategorien die Wahl zwischen gleichverteilten Zeiteinheiten für alle Kategorien oder längeren Zeiteinheiten für Kategorien, die er höher priorisiert hat. Hat der Nutzer zum Beispiel drei Kategorien ausgewählt und eine Nachrichtendauer von 30 Minuten angewählt, so erhält er bei gleichverteilter Gewichtung aus jeder Hauptkategorie Artikel mit einer Länge von insgesamt 10 Minuten. Für die konkrete Implementierung des priorisierenden Gewichtungsalgorithmus kommen verschiedene Funktionen in Frage. Beispiel: Hat der Nutzer, bei gleicher Kategorienwahl und Dauer die priorisierte Verteilung gewählt, haben die Artikel aus der obersten Kategorie z.B. eine Länge von 15 Minuten, jene aus der zweiten Kategorie dauern z.B. 10 Minuten und die aus der untersten Kategorie haben eine Dauer von z.B. 5 Minuten.

10. Anwendungsfall: PIN-Eingabe

Akteur: Nutzer
Übergeordneter Anwendungsfall: Profil erstellen
Beschreibung: Zur Sicherung des eigenen Profils gegen Überschreiben oder nicht autorisierten Zugriff wird jedem Profil eine PIN hinterlegt. Die Eingabe der PIN ist dabei nicht optional.

11. Anwendungsfall: Profil anpassen

Akteur: Nutzer

Vorbedingung: MACS MyNews ist verfügbar, Fahrzeug fährt langsamer als 5 km/h, zur Änderung eines SIM-gekoppelten Profils: Mobiltelefon verfügbar

Nachbedingung: Das geänderte Profil ist im internen Speicher aktualisiert

Beschreibung: Um Änderungen am aktuellen Profil vorzunehmen, kann der Nutzer im Menü „Optionen" den Punkt „Profil anpassen" wählen. Die Einstellungsmöglichkeiten unterscheiden sich nicht von denen beim Erstellen eines Profils. Es ist nicht möglich, während der Fahrt Änderungen an einem Profil durchzuführen.

12. Anwendungsfall: Profil als Standardprofil hinterlegen

Akteur: Nutzer

Vorbedingung: MACS MyNews ist verfügbar, Fahrzeug fährt langsamer als 5 km/h, Nutzerprofil geladen, Mobiltelefon verfügbar

Beschreibung: Um ein persönliches Profil auch ohne Mobiltelefon nutzen zu können, bietet der Newsdienst dem Nutzer die Möglichkeit, sein Profil als Standardprofil im Fahrzeug zu hinterlegen. Dazu wählt der Nutzer im Optionenmenü die Funktion „als Standardprofil hinterlegen" aus. Der Newsdienst fragt nach einer Bestätigung für das Überschreiben des aktuellen Standardprofils und führt bei positiver Antwort den Befehl aus. Der Nutzer wird über den Erfolg der Aktion unterrichtet. Wählt man das Optionenmenü an, während man das Standardprofil nutzt, ist diese Funktion nicht verfügbar. Das Standardprofil ist öffentlich zugänglich, siehe Anwendungsfall 15.

13. Anwendungsfall: Profil löschen

Akteur: Nutzer

Vorbedingung: MACS MyNews ist verfügbar, Fahrzeug fährt langsamer als 5 km/h, zur Löschung eines SIM-gekoppelten Profils: Mobiltelefon verfügbar

Nachbedingung: Das geänderte Profil ist aus dem internen Speicher gelöscht

Beschreibung: Nicht mehr benötigte Profile sollen vom Nutzer aus dem System entfernt werden können. Bei SIM-gekoppelten Profilen dient das Mobiltelefon als Benutzeridentifikation; in den übrigen Fällen muss zum Löschen des Profils, die dem Profil zugewiesene PIN eingegeben werden.

Sonderfälle: Nur ein Benutzer in Administratorenrolle kann lokale Profile löschen, deren PIN vergessen wurde (Vorschlag: Vertragswerkstatt als „Admin“).

14. Anwendungsfall: Profil wechseln

Akteure: Fahrer, Nutzer

Vorbedingung: MACS MyNews ist verfügbar, Profile verfügbar

Nachbedingung: Gewünschtes Profil wurde geladen

Beschreibung: Um verschiedene Interessenlagen beim Benutzer des Newsdienstes abdecken zu können, ist es möglich, verschiedene Profile anzulegen. Der Fahrer oder Nutzer des Newsdienstes soll einfach zwischen den Profilen wechseln können, dies gilt auch für den Fall, dass beim Einstieg in den Dienst z.B. das Mobiltelefon nicht verfügbar war oder ein neues Profil vom Nutzer importiert wurde. Um das Profil zu wechseln, geht der Fahrer/Nutzer in das Optionenmenü und wählt dort den Punkt „Profil wechseln“ aus. Er bekommt daraufhin vom Newsdienst alle intern verfügbaren Profile präsentiert und wählt eines dieser Profile aus. Möchte er ein SIM-gekoppeltes Profil laden, muss entweder das zugehörige Mobiltelefon verfügbar sein oder die im Profil hinterlegte PIN eingegeben werden. Der Wechsel zum fahrzeuginternen Standardprofil erfordert die Eingabe der PIN nicht.

15. Anwendungsfall: Profil importieren

Akteur: Nutzer

Vorbedingung: MACS MyNews ist gestartet, Fahrzeug fährt langsamer als 5 km/h, externes Speichermedium verfügbar

Nachbedingung: Profil von externem Speichermedium in den internen Speicher übertragen

Beschreibung: Um Profile auch zwischen zwei Fahrzeugen austauschen oder Sicherheitskopien anlegen zu können, soll dem Nutzer die Möglichkeit gegeben werden, sein Profil auf einem externen Medium zu speichern. Als externe Medien sind vor allem jene interessant, die schon über eine passende Schnittstelle im MMI verfügen (z.B. SD-Card) oder einfach angeschlossen werden können (z.B. PDA über Bluetooth). Eine gesonderte Sicherheitsüberprüfung fällt beim Import eines Profils weg, der Besitzer des Speichermediums wird als Profileigentümer gesehen.

Sonderfälle:

- Ist ein gelistetes Profil nicht (mehr) verfügbar, so gibt der Newsdienst eine Fehlermeldung aus.
- Ist kein interner Speicherplatz mehr frei, so wird der Nutzer aufgefordert, zuerst ein Profil aus dem Speicher zu löschen.

16. Anwendungsfall: Profil exportieren

Akteur: Nutzer

Vorbedingung: MACS MyNews ist gestartet, Fahrzeug fährt langsamer als 5 km/h, externes Speichermedium verfügbar

Nachbedingung: Profil von internem Speichermedium in den internen Speicher übertragen

Beschreibung: Da es möglich sein soll, Profile zwischen Fahrzeugen auszutauschen, soll dem Nutzer ermöglicht werden, seine Profile auf ein externes Speichermedium zu exportieren. Die Kriterien für das Medium sind offensichtlich dieselben wie beim Anwendungsfall „Profil importieren“. Beim Export eines Profils erfolgt zudem noch eine Sicherheitsüberprüfung, ob das eigene Profil exportiert wird.

Sonderfälle:

- Ist ein gelistetes Profil nicht (mehr) verfügbar, so gibt der Newsdienst eine Fehlermeldung aus.
- Ist ein gelistetes externes Medium nicht (mehr) verfügbar, so gibt der Newsdienst ebenfalls eine Fehlermeldung aus.

3.2.3 Analyse und Systementwurf

Nach der Erhebung der Anforderungen werden diese formalisiert in die Lösungsdomäne übertragen. Dazu werden als erstes das dynamische Verhalten der Teilfunktionen von MACS MyNews anhand der Nutzungsszenarien aufgestellt, anschließend werden zu hinterlegende Daten bestimmt, MyNews in Teilkomponenten zerlegt und deren Laufzeitverhalten in einem Kontrollfluss dargestellt.

1. Dynamisches Verhalten

Im folgenden Kapitel werden die komplexeren Szenarien von MACS MyNews genauer erläutert. Einfache Funktionsabläufe lassen sich aus den Anwendungsfällen der Anforderungserhebung direkt ablesen.

Zum *Einstieg in den Dienst* (vgl. Kapitel 3.2.2, 1. Anwendungsfall) wählt der Fahrer oder Nutzer aus einer Liste der verfügbaren Dienste den MACS MyNews Dienst durch Benutzung des haptischen Interface oder per Sprachbefehl. Je nach Verfügbarkeit von Profilen und eines Mobiltelefons (SIM-gekoppelte Profile) unterscheidet sich das Verhalten von MACS MyNews. Ist kein Profil vorhanden, kommt der Benutzer sofort zur Direktanwahl der gewünschten Nachricht (vgl. Kapitel 3.2.2, 2. Anwendungsfall). Steht kein Mobiltelefon oder kein mit dem vorhandenen Mobiltelefon gekoppeltes Profil zur Verfügung startet der Dienst mit dem im Fahrzeug hinterlegten Standardprofil. Ist ein mit dem verfügbaren Mobiltelefon gekoppeltes Profil vorhanden, startet der Dienst mit diesem Profil. Bei jeder der drei Möglichkeiten erfolgt eine Ansage durch den Dienst ob und welches Profil benutzt wird; dies dient als indirekter Hinweis für den Benutzer auf mögliche Fehlerquellen, z.B. keine Verbindung zum Mobiltelefon.

Der Nutzer von MACS MyNews hat die Möglichkeit vor Fahrtantritt ein persönliches *Profil zu erstellen* (vgl. 3.2.2, 5. Anwendungsfall). Hierzu wählt der Nutzer den entsprechenden Punkt aus dem MACS MyNews Menü aus. Je nach Verfügbarkeit eines Mobiltelefones wird ein SIM-gekoppeltes Profil erstellt. Dem Nutzer werden zunächst die verfügbaren Themengebiete präsentiert, er kann daraufhin ganze Kategorien oder einzelne Unterkategorien an- und abwählen (vgl. 3.2.2, 6. Anwendungsfall). Ist der Nutzer mit der Wahl der Kategorien fertig, kann er die Reihenfolge der Hauptkategorien angeben, in der er die Nachrichten hören möchte (vgl. 3.2.2, 7. Anwendungsfall). Dem Nutzer steht dann die Möglichkeit offen, eine gewünschte Newsdauer (vgl. 3.2.2, 8. Anwendungsfall) sowie eine Priorisierungsfunktion (vgl. 3.2.2, 9. Anwendungsfall) anzugeben. Um den Zugriff auf sein Profil auch ohne Authentifizierung durch ein Mobiltelefon zu ermöglichen, wählt der Nutzer nun eine PIN aus (vgl. 3.2.2, 10. Anwendungsfall) und benennt sein Profil. MACS MyNews speichert das Profil und präsentiert dem Nutzer die eben gewählten Inhalte.

Bei der Auswahl der Nachrichtenkategorien (vgl. 3.2.2, 6. Anwendungsfall) werden dem Nutzer alle verfügbaren Hauptkategorien präsentiert. Er kann die Auswahl an dieser Stelle wieder beenden, oder eine Hauptkategorie auswählen. War bislang keine Unterkategorie der Hauptkategorie gewählt, sind nun alle selektiert; waren bereits Unterkategorien gewählt, ändert sich daran nichts. Und beiden Fällen wird eine Liste aller verfügbaren Unterkategorien der aktuellen Hauptkategorie angezeigt. Der Nutzer kann einzelne Unterkategorien an- und abwählen oder zurück zur Anzeige der Hauptkategorien wechseln.

Die *Festlegung der Kategorienreihenfolge* (vgl. 3.2.2, 7. Anwendungsfall) erlaubt dem Nutzer des Systems die Reihenfolge der Darstellung der Inhalte gestaffelt nach Hauptkategorien vorzunehmen. Dazu wird wiederum die Liste aller Hauptkategorien aus seinem Profil dargestellt; der Nutzer kann nun eine der Hauptkategorien auswählen. Nach Selektion einer Hauptkategorie kann diese beliebig in der Liste verschoben werden. Wird eine Kategorie auf der obersten Position nach oben verschoben, wird sie an das Listenende gestellt; analog dazu wird eine Kategorie auf der untersten Position beim Verschieben nach unten auf die erste Position gesetzt.

Zur Auswahl der gewünschten *Dauer der Nachrichten* (vgl. 3.2.2, 8. Anwendungsfall) wird dem Nutzer eine Liste fest vorgegebener Zeitspannen vorgegeben, aus der er ein Element wählen kann. Eines der Elemente ist dabei eine „freie Eingabe“, bei der dem Nutzer in einem weiteren Dialog die direkte Eingabe der gewünschten Newsdauer erlaubt wird. Einzige Einschränkung dabei ist, dass die eingegebene ganze Zahl größer 0 ist.

Wählen Fahrer oder Nutzer die Option „*Profilwechsel*“ (vgl. 3.2.2, 14. Anwendungsfall), so werden zunächst alle Profile aus dem internen Speicher aufgelistet. Möchte der Fahrer oder Nutzer ein nicht SIM-gekoppeltes Profil laden, oder ist das mit dem Profil gekoppelte Mobiltelefon nicht verfügbar, wird er aufgefordert, die zum Profil passende PIN einzugeben. Informationen für Bezahldienste via Mobiltelefon werden in beiden Fällen nicht geladen. Wählt er ein SIM-gekoppeltes Profil und ist das Mobiltelefon vorhanden, wird das Profil sofort geladen.

2. Datenhaltung

MACS MyNews hält zwei Arten von Informationen bereit. Zum einen Nachrichteninhalte, die über eine breitbandige Funktechnologie in das Fahrzeug übertragen worden sind und Profilinformationen von Dienstenutzern, die entweder im Fahrzeug erstellt oder importiert wurden.

Das *Datenformat der Nachrichten* soll sich an verfügbaren Standards orientieren. Als minimale Anforderung gilt die Verwendung eines standardisierten Formates zur Darstellung der Inhalte, wünschenswert ist allerdings ein Format bei dem Syntax wie Semantik standardisiert sind. Ein Beispiel hierfür ist das von der Fraunhofer Gesellschaft entwickelte Newsservice Journaline.[78] Lediglich MACS MyNews hat Zugriff auf die Nachrichten; zur Absicherung von Bezahlinhalten sind diese verschlüsselt zu hinterlegen.

Nutzerpräferenzen werden in MACS MyNews als *Profilinformation* hinterlegt. Zu den hinterlegten Informationen zählen u.a. ein Profilname, eine Auswahl von Kategorien (vgl. 3.2.2, 6. Anwendungsfall) die gewünschte Dauer der Nachrichten (vgl. 3.2.2, 8. Anwendungsfall) sowie die Reihenfolge und Priorität der Kategorien (vgl. 3.2.2, 7. und 9. Anwendungsfall). Optional lassen sich Informationen zu Abonnements, Schlüssel für Bezahlinhalte u.ä. bei an ein Handy gekoppelten Profilen hinterlegen. Profile sind nur für MACS MyNews sichtbar und werden verschlüsselt gespeichert. Um die Informationen zu dechiffrieren, muss das gekoppelte Mobiltelefon verfügbar sein bzw. die hinterlegte PIN eingegeben werden (vgl. 3.2.2, 10. Anwendungsfall).

3. MyNews Teilkomponenten

Der MACS MyNews Dienst nutzt die in Kapitel 3.1 beschriebene Diensteplattform als Infrastrukturframework. Das bedeutet konkret, dass bei der Umsetzung der MACS MyNews Teilsysteme die von der Diensteplattform bereitgestellten Schnittstellen genutzt werden.

---

[78] Vgl. Zink (2005).

3.2.4 Implementierung

Durch die Umsetzung des prototypisch realisierten Application Frameworks (vgl. Kapitel 3.1) stehen zentrale Komponenten für die Umsetzung mobiler Anwendungen im Fahrzeug zur Verfügung. MACS MyNews wurde anhand dieses komponentenbasierten Ansatzes umgesetzt (vgl. Abbildung 19). Die Anwendung MyNews (dunkles Paket in Abbildung 19) selbst besteht also lediglich aus Funktionen zur Verwaltung von Nachrichten und Profilen sowie Routinen zur Erstellung von Abspiellisten. Übrige Funktionen (helle Pakete in Abbildung 19) sowie deren hardwarenahe Fahrzeugintegration (Pfeile in Abbildung 19) werden von der MACS Diensteplattform bereitgestellt.

Abbildung 19 Ausschnitt des MACS MyNews Pilotsystems

Da es sich bei MACS MyNews um einen diagonalen Prototypen handelt, wurde auf einige Anforderungen verzichtet. Von zentraler Bedeutung für die Evaluierbarkeit der Umsetzung, vor allem in Bezug auf ihr Gefährdungspotenzial, ist eine möglichst nahtlose Fahrzeugintegration (vgl. Kapitel 3.1). Für den Dienst bedeutet dies eine möglichst gute Abbildung des Bedienkonzeptes des Fahrzeugherstellers im Automobil. Weniger von Bedeutung hingegen ist ein komplett umgesetztes Profilmanagement. Präferenzen derart zu hinterlegen ist kein neuer Gedanke;[79] mehrere gleichzeitig im System hinterlegte Profile sind weder für Showcases noch für die Evaluation der Fahrsicherheit nötig.

[79] Zu den ältesten verfügbaren Angeboten im Bereich der personalisierbaren Nachrichten zählt AvantGo (http://www.avantgo.com)

Die bedeutendste Abweichung für MACS MyNews erfolgt in Bezug auf eine Funktion der Diensteplattform. Bei der prototypischen Umsetzung von MACS gibt es keine automatische Erkennung der Fahrsituation und damit keine automatische Steuerung der Art und Menge an Informationen, die dem Fahrer präsentiert werden. Vielmehr ist es möglich per Tastendruck zwischen dem „Fahrmodus“ und dem „Standbetrieb“ zu wechseln. Dies ermöglicht dem begleitenden Testleiter während der durchgeführten Fahrversuche, jederzeit das gewünschte Szenario herzustellen und damit für nachvollziehbare Ergebnisse auch zwischen verschiedenen Testfahrten zu sorgen.

Die zeitliche Ablaufsteuerung und die Nutzerinteraktion erfolgen analog zu der in der Anforderungserhebung festgelegten Nutzungssituation. Nach dem Start befindet sich der Prototyp im Zustand „News abspielen“ und beginnt automatisch mit dem Vorlesen von Nachrichten. Die weiteren beiden Zustände sind „Optionen anzeigen“ und „Pausiert“. Aus sämtlichen Zuständen lässt sich der Prototyp beenden.

Im Zustand „News abspielen“ wir der Text der aktuell vorgelesenen Nachricht angezeigt. Reicht der Platz auf dem Monitor zur Anzeige nicht aus, so kann im Text durch Drehen des Dreh-/Drückstellers gescrollt werden. Die in diesem Zustand möglichen Aktionen sind „Vor“ und „Zurück“ durch das Drücken der Vor- und Zurück-Taste bzw. die Sprachbefehle „Weiter“ und „Zurück“ zum Springen zwischen Nachrichten, sowie die Übergänge in die Zustände „Optionen anzeigen“ durch das Drücken des Softkeys „Optionen“ bzw. den Sprachbefehl „Optionen“ und in den Zustand „Pausiert“ durch das Drücken des Dreh-/Drückstellers bzw. den Sprachbefehl „Pause“. Ist eine Nachricht fertig vorgelesen, so wird automatisch zur nächsten gewechselt. Der Zustand „Pausiert“ wird durch erneutes Drücken des Dreh-/Drückstellers durch den Übergang in „News abspielen“ bzw. den Sprachbefehl „Weiter“ verlassen. In diesem Zustand ist es wie im Zustand „News abspielen“ möglich, zwischen Nachrichten zu springen. Beim Wechsel in den Zustand „Optionen anzeigen“ wird das Vorlesen der Nachrichten angehalten und eine Liste aktivierbarer Rubriken angezeigt, deren Nachrichten bei Aktivierung im Zustand „News abspielen“ vorgelesen werden. Rubriken werden durch das Drücken des Dreh-/Drückstellers bzw. durch den Sprachbefehl „Aktivieren“ bzw. „Deaktivieren“ aktiviert bzw. deaktiviert, durch das Drehen dessen bzw. die Sprachbefehle „Nach unten“ bzw. „Nach oben“ wird zwischen den Rubriken gewech-

selt. Durch erneutes Drücken des Softkeys „Optionen“ bzw. dem Sprachbefehl „Zurück“ wird in den Zustand „News abspielen“ gewechselt.

Abbildung 20 Komponentenübersicht MACS MyNews Pilotsystem

Der Prototyp des MACS MyNews Dienstes ist so designed, dass die komponentenbasierte Struktur der MACS Diensteplattform optimal ausgenutzt wird (vgl. Abbildung 20). Die MyNews Komponente „Nachrichten“ ist für das Abrufen, Speichern und die Bereitstellung der Nachrichten verantwortlich. Über den Daten- und Persistenzmanager der Diensteplattform werden die Nachrichten von den zur Verfügung stellenden Dienste-anbietern, sei es beispielsweise per GSM, UMTS, WLAN oder DAB, abgerufen und in Java-Objekte abgebildet. Dadurch ist es möglich sie anderen Komponenten von MACS MyNews zur Verfügung zu stellen. Durch einheitliche Schnittstellen der Nachrichten-Objekte können beliebige Nachrichtendienste eingebunden werden, indem für diese Dienste Klassen zum Abfragen und Konvertieren entwickelt und eingebunden werden, analog der Entwicklung und Anbindung von weiteren Ein- oder Ausgabegeräten für die Ein- bzw. Ausgabemanager. Für die persistente Speicherung im Automobil wird, wie bei der Abfrage neuer Nachrichten, der Daten- und Persistenzmanager verwendet. So ist die Benutzung von MACS MyNews auch möglich, wenn keine Verbindung zum Diensteanbieter besteht.

Aufgabe der Komponente „Profil“ ist die Verwaltung und Speicherung der persönlichen Einstellungen von MACS MyNews. Für jeden Benutzer werden die für die Anwendungsfälle 6 bis 10 (vgl. Kapitel 3.2.2) benötigten Informationen in seinem persönlichen Profil abgelegt. Dazu zählen unter anderem die abonnierten Nachrichten-Kategorien, die Reihenfolge und Gewichtung der jeweiligen Kategorien, die gewünschte Nachrichtendauer sowie eine PIN, und optional Mobiltelefoninformationen aus dem SIM Access Profile zur Sicherung des Profils. Zur persistenten Speicherung der unterschiedlichen Profile im Automobil greift diese Komponente auf der Daten- und Persistenzmanager der Basisdienste zurück. Als zusätzliche Absicherung vor unbefugtem Zugriff dienen die in den einzelnen Profilen hinterlegten Besitzerinformationen (SIM Access Profile und PIN).

Die Komponente HCI steuert die grafische Mensch-Maschine-Schnittstelle. Je nach bereits beschriebenem Zustand des Dienstes konfiguriert sie den Ausgabemanager und definiert die Bildschirmanzeige. Standardmäßig werden alle Nachrichten jederzeit vorgelesen, da hier ein geringes Ablenkungspotenzial besteht. Ist eine Nachricht komplett vorgelesen, wird automatisch zur nächsten Nachricht in der Playliste des Profils und fungiert so als „Taktgeber“ für die Anwendung. Parallel zur Ausgabe der synthetisierten Texte über die Lautsprecher erfolgt die Darstellung der Nachrichtentexte im Display sofern es die Verkehrssituation zulässt, d.h. wenn das Auto steht. Analog dazu sind sämtliche Multimediainhalte wie Videos oder Bilder ebenfalls während der Fahrt nicht verfügbar. Diese Funktionalität wird allerdings nicht direkt von der HCI Komponente bereitgestellt, sondern vom Application Safety Manager der Diensteplattform. Auf Eingabeseite fungiert die HCI Komponente lediglich als Aggregator der Informationen, die vom Benutzer zurück zum System fließen. Sämtliche Eingabegeräte können jederzeit benutzt werden, da eine fehlende Reaktion eines Eingabegerätes während der Fahrt auf den Fahrer eher irritierend wirken kann.

Der Controller ist die zentrale Komponente von MACS MyNews und koordiniert Start, Betrieb und Abschalten des Dienstes. Die Kommunikation der Komponenten innerhalb des Dienstes läuft über die Controllerkomponente, die wiederum das OSGi Framework benutzt, um mit Basisdiensten der MACS Diensteplattform zu kommunizieren. Benutzereingaben des Inputmanagers werden von der HCI Komponente ent-

gegen genommen und zusammen mit dem aktuellen Zustand der Komponenten Profil und Nachrichten zum aktuellen Zustand verarbeitet. Je nach Fahrsituation wird dieser Zustand dann über die HCI Komponente an den Fahrer ausgegeben und andere betroffenen Komponenten aktualisiert, z.B. die Playlist verändert.

3.2.5 Test und Evaluation

Bei den, auf die einzelnen Phasen abgestimmten, Tests während der Entwicklung von MACS MyNews wird auf die Methoden zurückgegriffen, die in Kapitel 2.4.4 beschrieben und bereits bei der Entwicklung der MACS Diensteplattform (Kapitel 3.1) benutzt wurden,. Dabei wird für jede einzelne Iteration des Spiralmodells (vgl. Abbildung 2) ein kompletter Satz Tests durchgeführt, unterteilt in die Phasen der Implementierung, der Analyse und des Systementwurfes sowie der Anforderungserhebung.

Die Tests der Phasen der Implementierung sowie der Analyse und des Systementwurfes unterscheiden sich dabei nicht von jenen Tests, die bereits im Rahmen der Entwicklung der Diensteplattform durchgeführt wurden. Durch den komponentenorientierten und iterativen Ansatz der Entwicklung ist es auch hier möglich, einzelne Klassen für sich zu testen und anschließend Teilkomponenten zunächst als Simulation bereitzustellen, um Integrationstests durchführen zu können.

Die gesammelten Anforderungen an MACS MyNews wurden mit Hilfe von *Systemtests*, veranschaulicht durch GUI Mockups und einfache Pilotsysteme, mehrfach in Expertenrunden abgestimmt. Dabei wurden neben technischen Aspekten unter anderem auch Annahmen für den Businesscase angepasst oder Anforderungen aus dem Businesscase übernommen.

Mit Hilfe von Pilotsystemen wurden Akzeptanztests durchgeführt (vgl. EVAL) und die Usability des Dienstes während der Fahrt überprüft (vgl. SAFE). Das so erhaltene Feedback von externen Experten und Kunden wurde jeweils in der nächsten Iteration in den prototypisch umgesetzten Dienst MACS MyNews übernommen.

4. Vorbereitung der Prototypenübernahme in die Serie

4.1 Voraussetzungen für die Einführung einer neuen Funktion in die Serie

Mobile Dienste müssen verschiedenen Kriterien erfüllen, um für die Serie nutzbar zu sein: der Dienst muss einen klaren Nutzen für den Fahrer haben, der Business Case für den Dienst muss positiv sein, die technische Umsetzung soll auf technischen Standards beruhen und die Umsetzung im Fahrzeug mit vertretbarem Aufwand realisierbar sein.

4.1.1 Fahrzeug

Für die Erbringung von Telematikdiensten werden geeignete Ressourcen vorausgesetzt. Diese Ressourcen betreffen die Ausstattung des Fahrzeugs, das eine geeignete Datenschnittstelle, so zum Beispiel einen ausreichend performanten Rechner mit relevanter Software. besitzt. Zudem wird eine Schnittstelle zum Fahrer benötigt, die Interaktion erlaubt und Feedback an den Fahrer geben kann. Bei Audi wird bei Oberklassefahrzeugen das MMI Bedienkonzept mit einem abgesetzten Bedienteil und separatem Bildschirm verwendet.

In der nächsten Generation von Infotainmentsystemen bei Audi wird zudem eine Spracheingabe und -ausgabe verfügbar sein, die dann auch Diensten im Fahrzeug zur Verfügung steht. Die Funktion Texte in Sprache umzuwandeln wird bei dem beschriebenen MACS MyNews Dienst vorausgesetzt. Die Sprachbedienung dient der Nutzung des Dienstes mit möglichst geringer Ablenkung auch während der Fahrt, wie im Berichtsteil zur Sicherheit und den Fahrversuchen beschrieben.

4.1.2 Software

Die Software in zukünftigen Automobilen wird modular aufgebaut und erweiterbar sein, so dass z.B. dieser Dienst als Komponente integriert bzw. nachträglich ergänzt, bzw. freigeschaltet werden kann. Hierfür müssen die einzelnen Teilkomponenten und die Schnittstellen sauber beschrieben und offen verfügbar sein. Die Basiskompo-

nenten der MACS Diensteplattform sind diesen Anforderungen entsprechend definiert und umgesetzt worden. Die Übernahme von Software in ein Serienprodukt bedingt zusätzliche Tests sowie einen serienkonformen Integrationsstand.

Für die umgesetzten Prototypen gilt als wichtigster Schritt in Richtung Serie das Oberflächendesign (GUI Design) auf Basis von Serienentwicklungswerkzeugen („Guide“ von 3 Soft), nachdem die Einbindung in ein definiertes Gesamtnutzungskonzept erfolgt ist.

4.1.3 Realisierungsoptionen für die Datenübertragung

Bei der Frage, wie die Daten ins Fahrzeug kommen, wurden im Laufe des Projekts zwei prinzipiell unterschiedliche Wege untersucht:

- Per Mobilfunk, also in einer eins zu eins Beziehung zwischen Fahrer/Fahrzeug und einer Zentrale.
- Per Rundfunk, also durch eine Ausstrahlung, bei der alle Teilnehmer dieselben Informationen empfangen können.

Aus Automotive-Sicht konnte eine Mobilfunkintegration im Fahrzeug keine positiven Ergebnisbeiträge generieren und scheidet daher auch in Zukunft als erste Wahl bei der Verbindung zum Kunden aus. Audi hat den Vertrieb neuer Fahrzeuge mit Telematik (Notruf, Serviceruf und Dynamischer Navigation per SMS) beendet und bietet jetzt Navigationssysteme mit dynamischer Navigation über RDS-TMC an. Die Einführung von RDS-TMC ist ein erfolgreiches Beispiel für den gemeinsamen Ansatz zur Etablierung neuer Standards in der Verkehrstelematik durch die beteiligten Fahrzeughersteller und Zulieferer sowie die Radioanstalten. 10 Jahre nach Einführung (1997 wurde TMC von Bosch/Blaupunkt in einem erstem Endgerät am Markt eingeführt) ist TMC bereits fester Bestandteil von modernen Navigationssystemen in Deutschland und wird international weiter eingeführt, unter anderem in China.

Die Abstimmung zur Einführung des TMC Standards wurde durch das neu gegründete TMC Forum unter Federführung von ERTICO eingeführt.[80] Nachfolger des TMC-Standards, der auf UKW beruht, könnte TPEG werden, der über DAB ausgestrahlt wird. Eine Anpassung auf die speziellen Fahrzeuganforderungen, unter anderem die übertragene Datenmenge oder die Bündelung verschiedener Dienste, wird zurzeit im Projekt Mobile.Info erarbeitet.

4.2 In welchem Umfeld könnte der Dienst entstehen?

4.2.1 MACS MyNews – ein Mobile.Info assoziierter Dienst

Eine mögliche Variante für die Realisierung von MACS MyNews ist die Assoziation von MyNews in Mobile.Info. Kernpunkt von Mobile.Info ist die Spezifizierung einer Diensteplattform, in der verschiedene Inhalte unterschiedlicher Anbieter angeboten werden können. Eine Nutzungsbeschränkung einzelner Inhalte durch ein „Conditional Access System" ist ein wichtiger Aspekt von Mobile.Info, da sowohl private als auch öffentlich-rechtliche Anbieter die Plattform verwenden werden und die privaten Anbieter planen, kostenpflichtige Mehrwertdienste anzubieten. Dabei werden die Inhalte zentral durch die jeweiligen empfängerspezifischen Schlüssel verschlüsselt und ausgestrahlt; die Empfänger entschlüsseln die Inhalte und können diese dann nutzen.

MACS MyNews nutzt für die Übertragung kostenpflichtiger Inhalte dieselben Mechanismen des Conditional Access, das HECA (High Efficient Conditional Access) System des Fraunhofer IIS. Vorteil des HECA Verfahrens ist die zentrale Verwaltung der Berechtigungsinformationen und die auf die Nutzdaten angewendete Verschlüsselung. Die Verschlüsselung erfolgt mit schnell wechselnden „Kontrollwörtern", was eine Überwinden der Verschlüsselung durch „Brute Force" (Austesten aller denkbarer Möglichkeiten mit hohem Rechenaufwand) und die Verwendung von einmal gekaufter Berechtigung durch mehrere, dann unberechtigte, Personen erschwert.

[80] Vgl. ERTICO (2006).

HECA ist ein lizenzpflichtiges Verfahren des Fraunhofer Instituts Erlangen, das pro Freischaltung oder Aktivierung im Endgerät Kosten verursacht.

Die Schlüsselverwaltung für HECA oder andere Mechanismen des Conditional Access kann über verschiedene Medien erfolgen, wie z.B. per USB Stick, dynamisch per Mobiltelefon oder WLAN. Aus Produktionssicht ist ein zentrales Einbringen in der Produktion, in Verbindung mit einem Ausstattungsmerkmal für einen Fahrzeugtyp, die geeignetste Methode der Umsetzung. Ein Verlängern oder Ändern der Freischaltung kann dann beim Händler vor Ort erfolgen, so dass die Rückdokumentation, d.h. die zentrale Speicherung, wie die Fahrzeuge aufgebaut sind, jederzeit möglich ist.

4.2.2 Newsservice Journaline

Im Rahmen eines vom BMBF geförderten Forschungsprojektes, RadioMondo, wurde vom Fraunhofer Institut für integrierte Schaltungen (IIS) ein Nachrichtendienst für schmalbandige Broadcastsendesysteme entwickelt. Im Gegensatz zum Broadcast Website Dienst, wie er für den DAB Standard definiert wurde, ist dieser Dienst auch in Geräten mit kleiner Prozessorleistung und kleinem Display anwendbar. Entwickelt wurde der Newsservice Journaline (NSJ) unter Beteiligung von den Partnern Bosch, Sony, T-Systems, Atmel und Micronas.

Die wichtigsten Komponenten des Newsservice Journaline sind die folgenden vier Inhaltekategorien:

- Liste: zum Aufzählen von kurzen Punkten, automatische Updates der Inhalte
- Menü: zum hierarchischen Gliedern, automatische Updates
- Text mit Überschrift; falls Element aktualisiert wird, während der Nutzer liest, wird es ihm signalisiert
- Nur Überschrift, mit automatischen Updates

Diese Komponenten können jeweils für unterschiedliche Zwecke verwendet werden. Jedes Inhalteelement ist selbstbeschreibend und kann autark genutzt werden, so

dass sogar ein Empfänger mit minimalem Speicher, der genau ein Element mit 2KB aufnehmen kann, den Dienst nutzen könnte.

NewsService Journaline wurde zunächst im Rahmen der Softwareempfänger für DRM (Digital Radio Mondiale) umgesetzt und ist in DRM bereits im Serienbetrieb verfügbar (z.B. Deutsche Welle). Der Dienst ist für DAB identisch nutzbar und ist für DAB bereits in den Standard aufgenommen. Dies ist die Voraussetzung für einen einheitlichen Einsatz. Der Standard des NewsService Journaline bietet die Möglichkeit neue Funktionen zu ergänzen, die in separate Felder geschrieben werden (erweiterter Header) oder im Nutztext durch „Tags" markiert werden können. Die Abwärtskompatibilität ist dabei gewährleistet, da Empfänger, die diese Zusatzinfos nicht verarbeiten können, unbekannte neue Tags ignorieren.

4.3 Der Weg in die Serie

Vor einer Serienentscheidung müssen definierte Schritte durchlaufen werden. Die technische Produktbeschreibung, als Basis für spätere Umsetzung, und ein positiver Business-Case werden einem Steuergremium zur Entscheidung vorgelegt. Nach positivem Bescheid wird festgelegt, ob der Umfang intern oder von einer externen Firma umgesetzt wird. Die Entwicklung erfolgt basierend auf dem verbindlichen Produktprozess (PEP) des Automobilherstellers, beziehungsweise dem für Elektronikumfänge üblichen Prozess. Hier werden einzelne Phasen der Detailproduktspezifikation und Feinplanung durchlaufen; die Entwicklung endet mit Komponenten und Systemtests, sowie Abnahmefahrten, bei denen die Serientauglichkeit der neuen Anwendung abgeprüft wird.

Zu Beginn des Einführungsprozesses muss entschieden werden, in welchem Rahmen der neue Dienst entstehen soll (vgl. Kapitel 4.2). Im Folgenden werden zwei verschiedene Ansätze kurz umrissen: einerseits eine rein proprietäre Lösung des Automobilherstellers und andererseits eine Lösung auf Basis bereits verfügbarer Standards.

4.3.1 MACS MyNews als proprietäre Implementierung

Für den Automobilhersteller besteht zum Projektabschluss des MACS Projektes die Möglichkeit den MACS MyNews Dienst ohne weitere Veränderungen in den Integrationsprozess zu übernehmen. Das beinhaltet vor allem auch die für mobile Dienste entscheidenden Schnittstellen auf Senderseite und zum Endkunden hin. Die Nachrichten würden in einem für den Prototypen eigens entwickelten, und damit proprietären, Format auf Senderseite aggregiert und übertragen. Eigenschaften wie die Aktualisierungsrate, die Aufteilung der Kategorien sowie die (groben) Inhalte sind dabei vorgegeben.

Großer Vorteil dieses Vorgehens ist, dass keinerlei Lizenzkosten für die verwendeten Standards anfallen würden, ganz im Gegensatz zu einer Lösung wie sie in folgendem Kapitel beschrieben wird. Nachteilig ist, dass die für den Prototypen entwickelte Lösung als „Proof-of-Concept" gedacht ist und entsprechend umgesetzt wurde. Die Folge ist, dass für einen Regelbetrieb der Bedarf an Übertragungsbandbreite vergleichsweise hoch ist, da immer ein kompletter Satz Nachrichten übertragen werden und nicht einfach Teile aktualisiert werden können. Bislang ist auch keine Einschränkung des Datenzugriffs, wie er z.B. für bezahlte Dienste nötig wäre, im Protokoll vorgesehen.

4.3.2 MACS MyNews auf Basis Newsservice Journaline

Der NewsService Journaline ist ein auf die textuelle Ausgabe orientierter Dienst, der durch manuelle Interaktion vom Nutzer bedient wird. Das Lesen der Inhalte ist die einzige Ausgabemethode. Eine Personalisierung ist bis auf „Bookmarks", mit denen bestimmte Seiten markiert werden können, nicht vorgesehen. Weitere Möglichkeiten der Personalisierung durch Profile sind nichtgeplant.

Für MACS MyNews bietet der NewsService Journaline eine gute, bereits etablierte Basis zur Datenübertragung sowie eine nicht-proprietäre Schnittstelle zur Umsetzung von MyNews bei weiteren Automobilherstellern. Anforderungen die vor der Nutzung von Journaline erfüllt werden müssen, sind unter anderem die konsequente Nutzung der Erweiterungsmöglichkeiten des Newsservice Journaline, die Erweiterung der

Texte um erforderliche Sprachzusatzinformationen (in SSML) sowie die Definition der erforderlichen Menge an verwendbaren Sprachbefehlen in SSML, so dass verschiedene Endgeräte mit den selben Inhalten einen einheitlichen Dienst anbieten. Bisher existiert jedoch keine NewsService Journaline Implementierung für DAB, sie wird jedoch zurzeit durch das Fraunhofer-Institut in Kooperation mit AUDI umgesetzt.

5. Implikationen und Ausblick

Ergebnisse der Pilotierungsphase im Projekt MACS sind ein prototypisch realisierter Dienst, MACS MyNews, ein Entwicklungsprozess zur systematischen Diensteentwicklung im Fahrzeug sowie eine funktionsfähige Softwareintegration in ein Serienfahrzeug. Der MACS MyNews Prototyp ist als Schichtenarchitektur aufgebaut und ermöglicht durch die Verwendung offener Standards die Austauschbarkeit und Erweiterbarkeit bestehender Funktionen durch Dritte. Dabei ist die komponentenorientierte Umsetzung von MACS MyNews ein besonderer Vorteil: Änderungen sind aufgrund der Austauschbarkeit einzelner Module Zeit und Kosten sparend durchführbar. Der vorgestellte Entwicklungsprozess erlaubt ein effizientes Co-Design der Dienstleistung und deren technischer Umsetzung. Durch die vielen Evaluationsphasen und verschiedene Prototypenstände, auch im Fahrzeug, sind Kunden leicht in den Entwicklungsprozess mit einzubinden.

Die praktische Anwendung des Design Framework anhand des Vorgehensmodells bei der Entwicklung von MACS MyNews hat gezeigt, dass es möglich ist, rasch zielgerichtetes Nutzerfeedback zu erhalten. Durch das iterative Vorgehen ergeben sich eine Reihe verschiedener Prototypen die jeweils den aktuellen Schritt im Design Framework betrachten. Auf diese Weise ist es möglich, zu den identifizierten Hauptgesichtspunkten detaillierte Kundenbedürfnisse bzw. -wünsche zu erhalten. Das gilt in erster Linie für die Gestaltung der Benutzerschnittstelle im Fahrzeug, hier haben die Benutzer die Möglichkeit die Usability des neuen Dienstes zu beeinflussen. Dazu zählt auch die Ausprägung einzelner Servicemerkmale, also des Funktionsumfanges des Dienstes. Beide Gesichtspunkte nehmen wiederum Einfluss auf die Sicherheit der Nutzung in der Fahrsituation, die jederzeit gegeben sein muss. Durch die Angabe von Präferenzen, z.B. die präsentierten Inhalte betreffend, lassen sich Rückschlüsse

auf die technische Basis der Dienstleistung und benötigte Wertschöpfungspartner ziehen. So unterschieden sich beide deutlich, wenn bei den Inhalten lediglich zwischen „personalisiert“ (also bereits außerhalb des Fahrzeuges auf den Kunden zugeschnitten) und „personalisierbar“ (wird im Fahrzeug zusammengestellt) unterschieden wird.

Das Design Framework eignet sich, um für alle Beteiligten eine gemeinsame Sicht auf den Dienst zu ermöglichen. Dadurch, dass jeder einzelne Schritt gekapselt ist und konkrete Ergebnisse erarbeitet werden, kennen alle Beteiligten den aktuellen Stand und können aktiv bei Entscheidungen diskutieren. Nachteil hierbei ist die fehlende Parallelisierbarkeit der einzelnen Schritte. Es können z.B. nicht neue Technologien evaluiert und prototypisch umgesetzt werden, während eine frühere Version auf ihre Sicherheitsaspekte hin überprüft wird. In einer weiteren Version des Design Frameworks sollen entsprechend Prozesse, die einander nicht direkt beeinflussen, auch parallel bearbeitet werden können. Des Weiteren müssen die einzelnen Methoden in den Ebenen konkretisiert werden; idealerweise bekommen Entwickler zukünftig konkrete Methodenbaukästen und Checklisten, anhand derer sie den jeweiligen Entwicklungsstand vorantreiben und überprüfen können.

Die Erfahrungen aus der Umsetzung der unterschiedlichen Iterationen des Prototyps legen den Schluss nahe, dass die für den Entwicklungsprozess vorgestellte Plattform grundsätzlich als Abstraktionsschicht im Automobil geeignet ist. Zusätzlich erscheint eine Rapid-Prototyping-Umgebung sinnvoll, da es Diensteentwicklern so möglich wäre, Dienste schon in frühen Stadien im Fahrzeug zu zeigen. Dies führt nicht nur zu einer Verbesserung der (internen) Projektkoordination, zusätzlich wird die Kommunikation zwischen verschiedenen Fachgruppen erleichtert und Kundenkliniken ermöglicht. Mit Hilfe eines solchen Werkzeuges zur Erstellung von Diensten lassen sich rasch neue Dienste aus dem Forschungsradar (vgl. Abbildung 15) umsetzen, da auf eine bestehende Infrastruktur zurückgegriffen werden und der Schwerpunkt der Entwicklungsarbeit auf neuen Komponenten liegen kann.

Neben möglichen Szenarien, die eigentliche Arbeit an einer Prototypingumgebung voranzutreiben, ergeben sich weitere viel versprechende neue Forschungsfelder.

Dazu zählt vor allem die Mensch-Maschine-Schnittstelle im Fahrzeug. Es gilt durch eine Rapid-Prototyping Plattform nicht nur möglich bestehende Designs möglichst gut an die Kundenbedürfnisse anzupassen, sondern vielmehr neue Möglichkeiten der Interaktion auszuloten. Das trifft auch auf neuartige Dienste oder Informationen zu; so wird die aktuelle Hardwareplattform des Prototypen dazu benutzt, den potentiellen Kunden, aber auch den Entwicklern und Entscheidungsträgern im Konzern, Einblicke in die Abläufe der aktuellen Hybridfahrzeuge zu ermöglichen. So ist es dem Kunden möglich, die ansonsten im Alltag kaum sichtbare Innovation live über die Multimediageräte im Fahrzeug zu verfolgen. Neben den Kunden als Hauptansprechpartner ergeben sich allerdings noch weitere mögliche Anspruchsgruppen, insbesondere im After-Sales-Bereich. So könnten z.B. interaktive Erweiterungen wie Avatare im Stil von Assistenzsystemen zu bestehenden Anleitungen in der Werkstatt die Mitarbeiter bei der Fehlerdiagnose und -behebung unterstützen oder die Qualitätssicherung durchgeführter Wartungsaufgaben dokumentieren.

Eine weitere Funktion, die durch eine Entwicklungsplattform erst ermöglicht werden kann, ist das Benchmarking bzw. die Evaluation von bestimmten Diensten und Dienstabläufen. So ist es denkbar, dass die Nutzung von (nachimplementierten) Infotainmentsystemen im Fahrzeug über einen längeren Zeitraum und über eine größere Zahl von Nutzern erhoben wird. Aus den Ergebnissen ließen sich dann Rückschlüsse über Nutzungshäufigkeiten einzelner Komponenten ziehen und Benutzungspfade innerhalb von Komponenten analysieren und optimieren.

Als logische Konsequenz hat sich für die aktuelle Forschung in Zusammenarbeit mit dem bisherigen Praxispartner vor allem das Themenfeld der Erweiterung des Human-Machine-Interface (HMI) durch Avatare ergeben. Im Hinblick auf die neuartige Mensch-Maschine-Kommunikation wird dabei ein avatarbasiertes HMI geplant und für zunächst statische Inhalte (wie das Benutzerhandbuch) und später auch für dynamische Inhalte (wie u.U. MACS MyNews) umgesetzt. Dabei sollen die Möglichkeiten der neuen Technologien aus verschiedenen Blickwinkeln untersucht werden. Einerseits können so die technische Machbarkeit und die möglichen Szenarien für den Einsatz solcher Systeme aufgezeigt werden, andererseits spielt auch die wirt-

schaftliche Verwertung der neuen Möglichkeiten durch die neuartige Benutzerinteraktion eine wesentliche Rolle.

Literaturverzeichnis

Abbott (1983): Program design by informal English descriptions, in: Communications of the ACM, 26. Jg. (1983), Nr. 11, S. 882-894.

Aksit/Tekinerdogan/Marcelloni/Bergmans (1999): Deriving Frameworks from Domain Knowledge, in: Building Application Frameworks - Object-oriented Foundations of Framework Design, hrsg. v. Fayad et al., New York 1999, S. 169-198.

Alliance (2005): About the OSGi Service Platform, in: Technical Whitepaper, Jg. (2005), Nr. S.

Balzert (1998): Lehrbuch der Software-Technik: Software-Entwicklung, 2. Auflage, Heidelberg et al 1998.

Beck (2003): Test Driven Development - By Example, Reading 2003.

Boehm (1988a): A Spiral Model of Software Development and Ehancement, in: IEEE Computer, 21. Jg. (1988a), Nr. 5, S. 61-72.

Boehm (1988b): A Spiral Model of Software Development and Enhancement, in: IEEE Computer, 21. Jg. (1988b), Nr. 5, S. 61-72.

Booch (1991): Object-Oriented Design with Applications, Redwood City 1991.

Booch (1994): Object-Oriented Analysis and Design with Applications, 2. Auflage, Redwood City 1994.

Boone (1999): Harvesting Design, in: Building Application Frameworks - Object-Oriented Foundations of Framework Design hrsg. v. Fayad et al., New York 1999, S. 199-210.

Bortz/Döring (2002): Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler, Berlin
Heidelberg 2002.

Brügge/Dutoit (2000): Object-Oriented Software Engineering: Conquering Complex and Changing Systems, Upper Saddle River 2000.

Carroll (1995): Scenario-based design, New York 1995.

Coad/Yourdon (1991a): Object-Oriented Analysis, 2. Auflage, Englewood Cliffs 1991a.

Coad/Yourdon (1991b): Object-Oriented Design, Englewoods Cliffs 1991b.

Daigle/van Gulik/Iannella/Faltstrom (2002): RFC 3406: Uniform Resource Names (URN) Namespace Definition Mechanismus, 2002.

Ehmer (2002): Mobile Dienste im Auto - Die Perspektive der Automobilhersteller in: Mobile Kommunikation: Wertschöpfung, Technologies, neue Dienste, hrsg. v. Reichwald, Wiesbaden 2002, S. 459-472.

ERTICO (2006): TMC Forum, in: Jg. (2006), Nr. S.

Fayad/Schmidt/Johnson (1999): Building Application Frameworks - Object-Oriented Foundations of Framework Design, New York 1999.

Floyd (1983): A Systematic Look at Prototyping. Paper presented at the Approaches to Prototyping, Namur 1983.

Fowler (2006): Continous Integration, in: Jg. (2006), Nr. S.

Frost/Sullivan (2003): Customer Attitudes and Perceptions Towards Telematics in Passenger Vehicles Market, in: Jg. (2003), Nr. S.

Fuhr (2001): Die Telematik ist tot - es lebe die rollende Schnittstelle. Paper presented at the Euroforum Jahrestagung "Telematik", Bonn 2001.

Grams (1990): Denkfallen und Programmierfehler, Berlin 1990.

Hansen/Hardie/Masinter (2006): RFC 4395: Guidelines and Registration Procedures for New URI Schemes, in: Jg. (2006), Nr. S.

Hartmann (2004): Wo viel Licht ist, ist starker Schatten - Softwareentwicklung in der Automobilindustrie, in: Automatisierumgstechnik, 52. Jg. (2004), Nr. 8, S. 353-358.

Heinecke/Schnelle/Fennel/Bortolazzi/Lundh/Leflour/Maté/Nishikawa/Scharnhorst (2004): AUTomotive Open System ARchitecture - An Industry-Wide Initiative to Manage the Complexity of Emerging Automotive E/E-Architectures. Paper presented at the International Congress on Transportation Electronics, Detroit, in: Jg. (2004), Nr. S. 325-332.

Jacobson/Booch/Rumbaugh (1999): The Unified Software Development Process, Reading 1999.

Jacobson/Christerson/Jonsson/Övergaard (1992): Object-Oriented Software Engineering - A Use Case Driven Approach, Reading 1992.

Jeffries/Anderson/Hendrickson (2001): Extreme Programming installed, München 2001.

Johnson/Foote (1988): Designing reusable classes, in: Journal of Object-Oriented Programming, 1. Jg. (1988), Nr. 5, S. 22-35.

Meyer (1997): Object-Oriented Software Construction, 2. Auflage, Upper Saddle River 1997.

Mohan (2006): Driving down the Fast Lane: Increasing Automotive Opportunities the EMS Provider Way, in: Frost & Sullivan Market Insight, Jg. (2006), Nr. S.

Oesterreich (1999): Objekt-orientierte Softwareentwicklung - Analyse und Design mit der Unified Modeling Language, München 1999.

Palenchar (2002): OSGi Networks Ready to Roll, in: TWICE, Jg. (2002), Nr. S.

Royce (1970): Managing the development of large software systems. Paper presented at the International Conference on Software Engineering, Monterey, in: Jg. (1970), Nr. S. 328-338.

Rumbaugh/Blaha/Permerlani/Eddy/Lorensen (1991): Object-Oriented Modeling and Design, Englewood Cliffs 1991.

Shlaer/Mellor (1988): Object-Oriented Analysis Modeling the World in Data, Englewood Cliffs 1988.

Shlaer/Mellor (1992): Object Lifecycles Modeling the World in States, Englewood Cliffs 1992.

Stevens/Pooley (2000): UML - Softwareentwicklung mit Objekten und Komponenten, München 2000.

Vaha-Sipila (2000): RFC 2806: URLs for Telephone Calls, in: Jg. (2000), Nr. S.

Werder (2005): Verkehrstelematik als Element der Verkehrspolitik. Paper presented at the itsch, Olten, in: Jg. (2005), Nr. S.

Wirfs-Brock/Wilkerson/Wiener (1990): Designing Object-Oriented Software Englewood Cliffs 1990.

Wong (2001): Open Services Gateway Initiative: OSGi Links Devices and Clients, in: Electronic Design, Jg. (2001), Nr. S. 86.

Zink (2005): NewsService Journaline - Der Nachrichtenservice für den digitalen Rundfunk. Paper presented at the 17th International Scientific Conference Mittweida, Mittweida, in: Jg. (2005), Nr. S. 20-23.